Chapter 7

MULTIVALUED DEPENDENCIES, JOIN DEPENDENCIES AND FURTHER NORMAL FORMS

We saw in Chapter 6 that the presence of certain functional dependencies in a relation scheme means that the scheme can be decomposed to eliminate redundancy while preserving information. However, it is not necessary that an FD hold before such a decomposition may take place. Consider the instance of relation *service* in Table 7.1.

Table 7.1 The relation *service*.

service(FLIGHT	DAY-OF-WEEK	PLANE-TYPE)
106	Monday	747
106	Thursday	747
106	Monday	1011
106	Thursday	1011
204	Wednesday	707
204	Wednesday	727

A tuple $\langle f\ d\ p \rangle$ in relation *service* means that flight number f flies on day d and can use plane type p on that day. There is no functional dependency FLIGHT → DAY-OF-WEEK or FLIGHT → PLANE-TYPE in *service*, yet *service* decomposes losslessly onto FLIGHT DAY-OF-WEEK and FLIGHT PLANE-TYPE, as shown in Table 7.2.

Table 7.2 The relation *service* decomposes losslessly into the relations *servday* and *servtype*.

servday(FLIGHT	DAY OF WEEK)	*servtype*(FLIGHT	PLANE-TYPE)
106	Monday	106	747
106	Thursday	106	1011
204	Wednesday	204	707
		204	727

Now consider another instance of the relation *service*, as given in Table 7.3.

Table 7.3 A second instance of the relation *service*.

service(FLIGHT	DAY-OF-WEEK	PLANE-TYPE)
106	Monday	747
106	Thursday	747
106	Thursday	1011
204	Wednesday	707
204	Wednesday	727

If we decompose this instance of service onto FLIGHT DAY-OF-WEEK and FLIGHT PLANE-TYPE, we also get the projections shown in Table 7.2. Therefore, when we join the two projections, we do not get back our original relation.

7.1 MULTIVALUED DEPENDENCIES

What property of the first instance of *service* that the second instance lacks allows the lossless decomposition? In the first instance, if a certain plane type can be used for a flight on one day it flies, that plane type can be used on any day the flight flies. This property fails for the second instance of *service*, since flight 106 can use a 1011 on Thursday but not on Monday. The first instance of *service* should be decomposed, since once we know a flight number, DAY-OF-WEEK gives us no information about PLANE-TYPE, and vice versa.

We can state this property another way. If we have tuples $\langle f\ d\ p \rangle$ and $\langle f\ d'\ p' \rangle$ in relation *service*, then we must also have tuple $\langle f\ d'\ p \rangle$. We define this concept formally.

Definition 7.1 Let R be a relation scheme, let X and Y be disjoint subsets of R, and let $Z = R - (X\ Y)$. A relation $r(R)$ satisfies the *multivalued dependency* (MVD) $X \twoheadrightarrow Y$ if, for any two tuples t_1 and t_2 in r with $t_1(X) = t_2(X)$, there exists a tuple t_3 in r with $t_3(X) = t_1(X)$, $t_3(Y) = t_1(Y)$, and $t_3(Z) = t_2(Z)$.

The symmetry of t_1 and t_2 in this definition implies there is also a tuple t_4 in r with $t_4(X) = t_1(X)$, $t_4(Y) = t_2(Y)$ and $t_4(Z) = t_1(Z)$.

Example 7.1 The MVD FLIGHT $\twoheadrightarrow$ DAY-OF-WEEK holds on the instance of *service* in Table 7.1, but not on the instance in Table 7.3. The instance in Table 7.1 also satisfies the MVD FLIGHT $\twoheadrightarrow$ PLANE-TYPE.

It is not a coincidence that the instance of *service* in Table 7.1 satisfies two MVDs, as the following lemma shows.

Lemma 7.1 If relation r on scheme R satisfies the MVD $X \twoheadrightarrow Y$ and $Z = R - (X\,Y)$, then r satisfies $X \twoheadrightarrow Z$.

Proof: Left to the reader (see Exercise 7.2).

The definition of MVD requires that X and Y be disjoint in $X \twoheadrightarrow Y$. Suppose we remove this condition from the definition. Let relation $r(R)$ satisfy $X \twoheadrightarrow Y$ under the modified definition and let $Y' = Y - X$. Under either definition, r satisfies $X \twoheadrightarrow Y'$: Let $Z = R - (X\,Y) = R - (X\,Y')$. Let t_1 and t_2 be tuples in r with $t_1(X) = t_2(X)$. Since $X \twoheadrightarrow Y$, there must be a tuple t_3 in r with $t_3(X) = t_1(X)$, $t_3(Y) = t_1(Y)$, and $t_3(Z) = t_2(Z)$. If $t_3(Y) = t_1(Y)$, then $t_3(Y') = t_1(Y')$, since $Y' \subseteq Y$. So r satisfies $X \twoheadrightarrow Y'$.

Now suppose that X and Y are disjoint and relation $r(R)$ satisfies $X \twoheadrightarrow Y$. If $X' \subseteq X$, then $X \twoheadrightarrow Y X'$ under the modified definition of MVD: If tuples t_1 and t_2 are in r, and $t_1(X) = t_2(X)$, then there is a tuple t_3 in r with $t_3(X) = t_1(X)$, $t_3(Y) = t_1(Y)$, and $t_3(Z) = t_2(Z)$. It follows that $t_3(Y\,X') = t_1(Y\,X')$.

We adopt the modified definition in place of the original.

Example 7.2 The relation r shown below satisfies the MVD $A\,B \twoheadrightarrow B\,C$, hence it satisfies the MVD $A\,B \twoheadrightarrow C$.

$r(A$	B	C	$D)$
a	b	c	d
a	b	c'	d'
a	b	c	d'
a	b	c'	d
a	b'	c'	d
a'	b	c	d'

Let us investigate the meaning of the special cases $\emptyset \twoheadrightarrow Y$ and $X \twoheadrightarrow \emptyset$ for a relation $r(R)$. Recall our assumption that $t(\emptyset) = \lambda$ for any tuple t. Consider $\emptyset \twoheadrightarrow Y$. Let $Z = R - Y$. For any tuples t_1 and t_2 in r, $t_1(\emptyset) = t_2(\emptyset)$. If r satisfies $\emptyset \twoheadrightarrow Y$, there must be a tuple $t_3 \in r$ with $t_3(Y) = t_1(Y)$ and $t_3(Z) = t_2(Z)$. Therefore, r must be the cross product of the projections $\pi_Y(r)$ and $\pi_Z(r)$.

The MVD $X \twoheadrightarrow \emptyset$ is trivially satisfied by any relation on a scheme containing X.

7.2 PROPERTIES OF MULTIVALUED DEPENDENCIES

We have formalized the property that distinguishes the instances of the relation *service* in Tables 7.1 and 7.3. Let us see how MVDs are related to lossless decomposition.

Theorem 7.1 Let r be a relation on scheme R, and let X, Y, and Z be subsets of R such that $Z = R - (X\,Y)$. Relation r satisfies the MVD $X \twoheadrightarrow Y$ if the only if r decomposes losslessly onto the relation schemes $R_1 = X\,Y$ and $R_2 = X\,Z$.

Proof: Suppose the MVD holds. Let $r_1 = \pi_{R_1}(r)$ and $r_2 = \pi_{R_2}(r)$. Let t be a tuple in $r_1 \bowtie r_2$. There must be a tuple $t_1 \in r_1$ and a tuple $t_2 \in r_2$ such that $t(X) = t_1(X) = t_2(X)$, $t(Y) = t_1(Y)$, and $t(Z) = t_2(Z)$. Since r_1 and r_2 are projections of r, there must be tuples t_1' and t_2' in r with $t_1(X\,Y) = t_1'(X\,Y)$ and $t_2(X\,Z) = t_2'(X\,Z)$. The MVD $X \twoheadrightarrow Y$ implies that t must be in r, since r must contain a tuple t_3 with $t_3(X) = t_1'(X)$, $t_3(Y) = t_1'(Y)$, and $t_3(Z) = t_2'(Z)$, which is a description of t.

Suppose now that r decomposes losslessly onto R_1 and R_2. Let t_1 and t_2 be tuples in r such that $t_1(X) = t_2(X)$. Let r_1 and r_2 be defined as before. Relation r_1 contains a tuple $t_1' = t_1(X\,Y)$ and relation r_2 contains a tuple $t_2' = t_2(X\,Z)$. Since $r = r_1 \bowtie r_2$, r contains a tuple t such that $t(X\,Y) = t_1(X\,Y)$ and $t(X\,Z) = t_2(X\,Z)$. Tuple t is the result of joining t_1' and t_2'. Hence t_1 and t_2 cannot be used in a counterexample to $X \twoheadrightarrow Y$, hence r satisfies $X \twoheadrightarrow Y$.

Theorem 7.1 gives us a method to test if a relation $r(R)$ satisfies the MVD $X \twoheadrightarrow Y$. We project r onto $X\,Y$ and $X(R - XY)$, join the two projections, and test if the result is r. There is another method to test MVDs that does not require project and join, only some sorting and counting.

Let $Z = R - (X\,Y)$, $R_1 = X\,Y$, and $R_2 = X\,Z$. If

$$r_1 = \pi_{R_1}(r) \quad \text{and} \quad r_2 = \pi_{R_2}(r),$$

then $r_1 \bowtie r_2$ always contains r. For a given X-value x, suppose there are c_1 tuples in r_1 with X-value x and c_2 tuples in r_2 with X-value x. Let c be the number of tuples in r with X-value x. If $c = c_1 \cdot c_2$, for any X-value x, then $r = r_1 \bowtie r_2$ (see Exercise 7.4).

We define a function to assist us with our counting. The function $c_W[X = x]$ maps relations to non-negative integers as follows:

$$c_W[X = x](r) = |\pi_W(\sigma_{X=x}(r))|$$

Example 7.3 The value of $c_D[A\ B = a\ b](r)$ is 2 for the relation r in Example 7.2.

The function $c_W[X = x]$ counts the number of different W-values associated with a given X-value in a relation. The condition for the MVD $X \twoheadrightarrow Y$ we just discussed can be stated as

For any X-value x in r, $c_R[X = x](r) = c_{XY}[X = x](r) \cdot c_{XZ}[X = x](r)$.

Since $c_{WX}[X = x] = c_W[X = x]$, we can simplify this condition to

For any X-value x in r, $c_R[X = x](r) = c_Y[X = x](r) \cdot c_Z[X = x](r)$.

Example 7.4 For the relation r in Example 7.2, and the MVD $A\ B \twoheadrightarrow C$,

$$c_{ABCD}[A\ B = a\ b](r) = 4,$$
$$c_C[A\ B = a\ b](r) = 2, \text{ and}$$
$$c_D[A\ B = a\ b](r) = 2.$$

We see the condition is satisfied for the $(A\ B)$-value $a\ b$.

To test a relation $r(R)$ against the MVD $X \twoheadrightarrow Y$, first let $Z = R - (X\ Y)$. Next, sort the relation to bring equal X-values together. For each X-value, we count the number of tuples with the value, the number of different Y-values associated with the X-value, and the number of different Z-values associated with the X-value. Finally, we test if the first number is the product of the other two.

This test provides another definition of MVD (see Exercise 7.6).

Definition 7.2 Let r be a relation on scheme R, let X and Y be subsets of R, and let $Z = R - (X\ Y)$. Relation r satisfies the *multivalued dependency* $X \twoheadrightarrow Y$ if for every X-value x and Y-value y in r, such that xy appears in r,

$$c_Z[X = x](r) = c_Z[X\ Y = x\ y](r).$$

7.3 MULTIVALUED DEPENDENCIES AND FUNCTIONAL DEPENDENCIES

From Theorem 7.1 we can derive the following corollary.

Corollary Let r be a relation on scheme R and let X and Y be subsets of R. If r satisfies the FD $X \rightarrow Y$, then r satisfies the MVD $X \twoheadrightarrow Y$.

Proof From Exercise 6.4, we know that $X \rightarrow Y$ implies r decomposes losslessly onto $X\,Y$ and $X\,(R - (X\,Y))$. This result also follows directly from the counting definition of MVD.

Suppose we have a relation scheme R and a set of FDs F over R. We want to know which MVDs must hold in a relation $r(R)$ that satisfies F. From the last corollary, we know that if $X \rightarrow Y$ is in F^+, then r satisfies $X \twoheadrightarrow Y$, and, by Lemma 7.1, it follows that r satisfies $X \twoheadrightarrow R - (X\,Y)$. Are there any MVDs that will always hold on r that do not correspond directly to FDs? The answer is no.

Theorem 7.2 Let F be a set of FDs over R. Let X, Y, and Z be subsets of R, with $Z = R - (X\,Y)$. Let X^+ be the closure of X under F. If $Y \not\subseteq X^+$ and $Z \not\subseteq X^+$, then there is a relation $r(R)$ that satisfies F and does not satisfy the MVD $X \twoheadrightarrow Y$.

Proof The proof is similar to that of Theorem 4.1 on the completeness of the inference axioms for FDs. Assume $R = A_1 A_2 \cdots A_n$. We construct a relation $r(R)$ containing only two tuples, t_1 and t_2. Tuple t_1 is defined as

$$t_1(A_i) = a_i, \qquad 1 \leq i \leq n$$

and tuple t_2 is defined as

$$t_2(A_i) = \begin{cases} a_i & \text{if } A_i \in X^+ \\ b_i & \text{otherwise,} \end{cases} \qquad 1 \leq i \leq n.$$

By the proof of Theorem 4.1, r satisfies all the FDs in F. Since $Y \not\subseteq X^+$ and $Z \not\subseteq X^+$, Y must contain an attribute B_1 not in X^+, and Z must contain an attribute B_2 not in X^+. Thus, $t_2(B_1) = b_j$ and $t_2(B_2) = b_k$ for some j and k.

Since $X \subseteq X^+$, $t_1(X) = t_2(X)$. If r satisfies $X \twoheadrightarrow Y$, r must contain a tuple t_3 with $t_3(X) = t_1(X)$, $t_3(Y) = t_1(Y)$, and $t_3(Z) = t_2(Z)$. However, r has only two tuples, so $t_3 = t_1$ or $t_3 = t_2$. Suppose $t_3 = t_1$. Then $t_3(B_2) = t_1(B_2) \neq t_2(B_2)$, since $t_2(B_2)$ is b_k and t_1 is all a's, so $t_3(Z) \neq t_2(Z)$; a contradiction.

Similarly, we get a contradiction if we assume $t_3 = t_2$. Since $t_2\,(B_1) = b_j$ and t_1 is all a's, $t_3(B_1) = t_2(B_1) \neq t_1(B_1)$, so $t_3(Y) \neq t_1(Y)$. We must conclude that r does not satisfy the MVD $X \twoheadrightarrow Y$.

From Theorem 7.2 we see that the only MVDs implied by a set of FDs are those of the form $X \twoheadrightarrow Y$, where $Y \subseteq X^+$ or $R - (X\,Y) \subseteq X^+$.

Example 7.5 Let $R = A\,B\,C\,D\,E\,I$ and let $F = \{A \to B\,C,\ C \to D\}$. Then F implies $A \twoheadrightarrow B\,C\,D$ and $A \twoheadrightarrow C$, but F does not imply $A \twoheadrightarrow D\,E$.

7.4 INFERENCE AXIOMS FOR MULTIVALUED DEPENDENCIES

We have just seen exactly which MVDs are implied by a set of FDs. We now consider what MVDs are implied by a set of MVDs and what MVDs and FDs are implied by a set of MVDs and FDs.

7.4.1 Multivalued Dependencies Alone

The first six inference axioms below are analogs to the FD axioms with the same names, although only the first three have exactly the same statement. Axiom M7 has no FD counterpart.

Let r be a relation on scheme R and let W, X, Y, Z be subsets of R.

M1. Reflexivity
Relation r satisfies $X \twoheadrightarrow X$.

M2. Augmentation
If r satisfies $X \twoheadrightarrow Y$, then r satisfies $X\,Z \twoheadrightarrow Y$.

M3. Additivity
If r satisfies $X \twoheadrightarrow Y$ and $X \twoheadrightarrow Z$, then r satisfies $X \twoheadrightarrow Y\,Z$.

M4. Projectivity
If r satisfies $X \twoheadrightarrow Y$ and $X \twoheadrightarrow Z$, then r satisfies $X \twoheadrightarrow Y \cap Z$ and $X \twoheadrightarrow Y - Z$.

M5. Transitivity
If r satisfies $X \twoheadrightarrow Y$ and $Y \twoheadrightarrow Z$, then r satisfies $Y \twoheadrightarrow Z - Y$.

M6. Pseudotransitivity
If r satisfies $X \twoheadrightarrow Y$ and $Y\,W \twoheadrightarrow Z$, then r satisfies $X\,W \twoheadrightarrow Z - (Y\,W)$.

M7. Complementation
If r satisfies $X \twoheadrightarrow Y$ and $Z = R - (X\,Y)$, then r satisfies $X \twoheadrightarrow Z$.

Axioms M1 and M2 follow immediately from the first definition of MVD (see Exercise 7.8). Let us demonstrate the correctness of axiom M3. Let r contain tuples t_1 and t_2, with $t_1(X) = t_2(X)$. We must prove that r contains a tuple t such that

$$t(X) = t_1(X), \qquad t(YZ) = t_1(YZ), \quad \text{and} \quad t(U) = t_2(U),$$

where $U = R - (X\,YZ)$. Since r satisfies $X \twoheadrightarrow Y$, it must contain a tuple t_3 such that

$$t_3(X) = t_1(X), \qquad t_3(Y) = t_1(Y), \quad \text{and} \quad t_3(V) = t_2(V),$$

where $V = R - (X\,Y)$. Since r satisfies $X \twoheadrightarrow Z$, it must contain a tuple t_4 such that

$$t_4(X) = t_1(X), \qquad t_4(Z) = t_1(Z), \quad \text{and} \quad t_4(W) = t_3(W),$$

where $W = R - (X\,Z)$.

We claim $t = t_4$. Clearly $t(X) = t_4(X)$.

Also

$$t_4(Z) = t_1(Z) = t(Z), \text{ and}$$
$$t_4(Y \cap W) = t_3(Y \cap W) = t_1(Y \cap W) = t(Y \cap W), \text{ so}$$
$$t_4(YZ) = t(YZ).$$

Since $U \subseteq W \cap V$,

$$t_4(U) = t_3(U) = t_2(U) = t(U).$$

We have shown $t_4 = t$, since $R = X\,Y\,Z\,U$.

We know axiom M7 is correct from Lemma 7.1. We can use axioms M3 and M7 to prove the correctness of axiom M4. If r satisfies $X \twoheadrightarrow Y$ and $X \twoheadrightarrow Z$, then, by axiom M3, r satisfies $X \twoheadrightarrow YZ$. By axiom M7, r must also satisfy $X \twoheadrightarrow V$, where $V = R - (X\,YZ)$. Using M3 again, we know r satisfies $X \twoheadrightarrow VZ$. A final application of M7 yields $X \twoheadrightarrow R - (X\;V\;Z)$. Substituting and simplifying gives us

$$\begin{aligned}&R - (X\,V\,Z) = \\ &R - (X\{R - (X\,Y\,Z)\}Z) = \\ &R - (X\{R - Y\}Z) = \\ &Y - (X\,Z) = \\ &(Y - Z) - X.\end{aligned}$$

Therefore, r satisfies $X \twoheadrightarrow (Y - Z) - X$, which implies $X \twoheadrightarrow Y - Z$ by the discussion in Section 7.1.

From $X \twoheadrightarrow Y$ we get $X \twoheadrightarrow W$ by axiom M7, where $W = R - (X\,Y)$. Combining this with $X \twoheadrightarrow Y - Z$ using axiom M3 yields $X \twoheadrightarrow W(Y - Z)$. One more application of axiom M7 gives us $X \twoheadrightarrow R - (X\,W(Y - Z))$. Substituting, we get

$$\begin{aligned}&R - (W\,X(Y - Z)) = \\ &R - (X\{R - (X\,Y)\}(Y - Z)) = \\ &R - (X\{R - Y\}(Y - Z)) = \\ &Y - (X(Y - Z)) = \\ &(Y \cap Z) - X.\end{aligned}$$

Thus r satisfies $X \twoheadrightarrow (Y \cap Z) - X$ and hence r satisfies $X \twoheadrightarrow Y \cap Z$.

To prove the correctness of axiom M5, we first show that $X \twoheadrightarrow Y$ and $Y \twoheadrightarrow Z$ imply $X \twoheadrightarrow YZ$. Let $W = R - (X\,Y\,Z)$. We must show that if there are tuples t_1 and t_2 in r, with $t_1(X) = t_2(X)$, then r contains a tuple t such that

$$\begin{aligned}&t(X) = t_1(X), \\ &t(YZ) = t_1(YZ), \text{ and} \\ &t(W) = t_2(W).\end{aligned}$$

From $X \twoheadrightarrow Y$, we get a tuple t_3 such that

$$t_3(X) = t_1(X), \qquad t_3(Y) = t_1(Y), \quad \text{and} \quad t_3(V) = t_2(V),$$

where $V = R - (X\,Y)$. Using $Y \twoheadrightarrow Z$ we get a tuple t_4 such that

$$t_4(Y) = t_1(Y), \qquad t_4(Z) = t_1(Z), \quad \text{and} \quad t_4(U) = t_3(U),$$

where $U = R - (Y\,Z)$.

We know $t_4(X) = t_1(X)$, since there is only one possible value for each attribute $A \in X$. Clearly $t_4(YZ) = t_1(YZ)$. Since $W \subseteq U - X \subseteq V$, $t_4(W) =$

$t_2(W)$. Hence, t_4 is the tuple t we seek. We have shown r satisfies $X \twoheadrightarrow Y Z$. Using axiom M4 and $X \twoheadrightarrow Y$, we finally get $X \twoheadrightarrow Z - Y$.

Axiom M6 follows from the other axioms and is left as an exercise (see Exercise 7.10).

Example 7.6 Let $R = A\ B\ C\ D\ E$ and let $F = \{A \twoheadrightarrow B\ C,\ D\ E \twoheadrightarrow C\}$. From $A \twoheadrightarrow B\ C$ we get $A \twoheadrightarrow D\ E$ by complementation. Transitivity then gives us $A \twoheadrightarrow C$. Using augmentation we get $A\ D \twoheadrightarrow C$. Finally, applying complementation again yields $A\ D \twoheadrightarrow B\ E$. Therefore $F \models A\ D \twoheadrightarrow B\ E$. Below is a relation $r(A\ B\ C\ D\ E)$ that satisfies all of these MVDs.

$r(A$	B	C	D	$E)$
a	b	c	d	e
a'	b'	c'	d	e
a'	b'	c	d	e
a	b	c'	d	e
a''	b'	c'	d'	e

7.4.2 Functional and Multivalued Dependencies

We now turn our attention to the implications we can make when we have FDs and MVDs together. There are only two axioms for FDs and MVDs combined.

Let r be a relation on R and let W, X, Y, Z be subsets of R.

C1. Replication
If r satisfies $X \rightarrow Y$, then r satisfies $X \twoheadrightarrow Y$.

C2. Coalescence
If r satisfies $X \twoheadrightarrow Y$ and $Z \rightarrow W$, where $W \subseteq Y$ and $Y \cap Z = \emptyset$, then r satisfies $X \rightarrow W$.

Axiom C1 is a consequence of the corollary to Theorem 7.1. We prove the correctness of axiom C2. Let t_1 and t_2 be tuples in r with $t_1(X) = t_2(X)$. Since r satisfies $X \twoheadrightarrow Y$, there must be a tuple t in r such that

$$t(X) = t_1(X), \qquad t(Y) = t_1(Y), \quad \text{and} \quad t(V) = t_2(V),$$

where $V = R - (X\ Y)$. Since $Y \cap Z = \emptyset$, $Z \subseteq X\ V$, hence $t(Z) = t_2(Z)$. The FD $Z \rightarrow W$ means that $t(W) = t_2(W)$. However, $W \subseteq Y$, so $t_1(W) = t(W) = t_2(W)$, hence r satisfies $X \rightarrow W$.

Example 7.7 Let $R = A\ B\ C\ D\ E$ and let $F = \{A \twoheadrightarrow BC,\ D \rightarrow C\}$. Axiom C2 implies $F \models A \rightarrow C$. Below is a relation $r(A\ B\ C\ D\ E)$ that satisfies these FDs and MVDs.

$r(A$	B	C	D	$E)$
a	b	c'	d	e
a	b'	c'	d'	e'
a	b'	c'	d	e
a	b	c'	d'	e'

7.4.3 Completeness of the Axioms and Computing Implications

We shall only state the completeness results for inference axioms involving MVDs; we shall not prove them here.

Theorem 7.3 Inference axioms M1–M7 are complete for sets of MVDs.

Theorem 7.4 Inference axioms F1–F6, M1–M7, and C1 and C2 are complete for sets of FDs and MVDs.

As a consequence of Theorem 7.4, we see that a set of MVDs alone implies no FDs other than trivial ones; that is, FDs of the form $X \rightarrow Y$, where X contains Y. This observation follows from the form of the inference axioms. F1–F6 can only derive trivial FDs from trivial FDs; M1–M7 and C1 cannot derive any FDs; axiom C2 does not apply when the FD involved is trivial.

Axioms C1 and C2 are necessary. Without axiom C1, MVDs could not be derived from a set of only FDs. It is left as an exercise to find an example where axiom C2 derives an FD that could not be derived from axioms F1–F6 alone from a given set of FDs and MVDs (see Exercise 7.12).

We shall not develop a membership algorithm for MVDs or FDs and MVDs, although polynomial-time algorithms exist in both cases. We shall, however, discuss some of the concepts used in these algorithms, since these concepts help give a better picture of the dependency structure implied by a set of MVDs.

Definition 7.3 Given a collection of sets $\mathbf{S} = \{S_1, S_2, \ldots, S_p\}$, where $\mathbf{U} = S_1 \cup S_2 \cup \cdots \cup S_p$, the *minimal disjoint set basis* of $\mathbf{S}$ ($mdsb(\mathbf{S})$) is the partition $T_1, T_2, \ldots, T_q$ of $\mathbf{U}$ such that:

1. Every S_i is a union of some of the T_j's.
2. No partition of $\mathbf{U}$ with fewer cells has the first property.

The reader should take a moment to convince himself or herself that the *mdsb*(**S**) is unique as defined. The *mdsb*(**S**) is formed by grouping together elements in **U** that are contained in exactly the same set of S_i's.

Example 7.8 Let **S** = $\{A\ B\ C\ D,\ C\ D\ E,\ A\ E\}$. We have **U** = $A\ B\ C\ D\ E$ and *mdsb*(**S**) = $A,\ B,\ C\ D,\ E$.

Let F be a set of MVDs over R and let $X \subseteq R$. Define G as

$$G = \{Y | F \models X \twoheadrightarrow Y\}.$$

We claim $mdsb(G)$ is a subset of G. If there is a set Y_1 in G such that Y_1 contains attributes both in and out of some other set Y_2 in G, then, by axiom M4, there are sets $Y_3 = Y_1 - Y_2$ and $Y_4 = Y_1 \cap Y_2$ in G. $Mdsb(G)$ consists exactly of those nonempty sets of G that contain no other set of G as a subset. Note that if $X = A_1 A_2 \cdots A_n$, then $A_1, A_2, \ldots, A_n$ are all in $mdsb(G)$.

Definition 7.4 Let F, X, and G be as defined above. The *dependency basis of* X with respect to F is $mdsb(G)$ and is denoted $DEP(X)$. If $X = A_1 A_2 \cdots A_n$ and $DEP(X) = \{A_1, A_2, \ldots, A_n, Y_1, Y_2, \ldots, Y_m\}$, we write $X \twoheadrightarrow Y_1 | Y_2 | \cdots | Y_m$.

Example 7.9 Let $F = \{A \twoheadrightarrow BC,\ DE \twoheadrightarrow C\}$ be a set of MVDs over $ABCDE$. If $X = A$, then $G = \{A, BC, DE, C, BDE, B, BCDE, CDE\}$ and $DEP(A) = mdsb(G) = \{A,B,C,DE\}$.

We can recover all MVDs implied by F with X as the left side from $DEP(X)$. $F \models X \twoheadrightarrow Y$ if and only if Y is the union of some sets in $DEP(X)$. Y must be in G, so Y is the union of some sets in $DEP(X)$. In the other direction, by axiom M3, if $Y_1, Y_2, \ldots, Y_k$ are in $DEP(X)$, then $F \models X \twoheadrightarrow Y_1 Y_2 \cdots Y_k$.

The membership algorithm for MVDs tests if a set of MVDs implies an MVD $X \twoheadrightarrow Y$ by first computing $DEP(X)$ with respect to F and then checking if Y can be formed from sets in $DEP(X)$. The procedure for computing the dependency basis of X has three stages.

1. Find the set G of all sets Y such that the MVD $X \twoheadrightarrow Y$ follows from F by augmentation of complementation. That is, for any MVD $X' \twoheadrightarrow Y'$ in F where $X' \subseteq X$, add Y' and $R - (X'\ Y')$ to G. Also add A to G for every $A \in X$.
2. Let $DEP(X) = mdsb(G)$.
3. Look for an MVD $W \twoheadrightarrow Z$ that can be used to refine $DEP(X)$ with

transitivity. That is, let $Y_1, Y_2, \ldots, Y_k$ be sets in $DEP(X)$ such that $W \subseteq Y_1 Y_2 \cdots Y_k$. Let $Y = Y_1 Y_2 \cdots Y_k$. By augmentation, since $W \subseteq Y$, $F \models Y \twoheadrightarrow Z$. By transitivity, $X \twoheadrightarrow Z - Y$. If $Z - Y$ is the union of some sets in $DEP(X)$, we cannot refine $DEP(X)$. If not, let $DEP(X) = mdsb(DEP(X) \cup \{Z - Y\})$. If no MVD in F can be used to change $DEP(X)$, stop.

Example 7.10 Let $F = \{A \twoheadrightarrow B\ C,\ D\ E \twoheadrightarrow C\}$ be a set of MVDs over $A\ B\ C\ D\ E$. To compute $DEP(A)$, we first find $G = \{BC, DE, A\}$. We then set $DEP(A) = \{B\ C, D\ E, A\}$. We then use transitivity on $D\ E \twoheadrightarrow C$ to get $A \twoheadrightarrow C$ and refine $DEP(A)$ to $mdsb(\{B\ C, D E, A\} \cup C) = \{B, C, D E, A\}$. We can make no further refinement to $DEP(X)$.

We shall not attempt to prove the correctness of the procedure for computing $DEP(X)$. Observe, however, that the time complexity of the procedure is bounded by a polynomial in the size of F. $DEP(X)$ can contain at most $|R|$ sets, thus $DEP(X)$ can be refined at most $|R - X| - 1$ times in step 3. (Any attribute in X is in $DEP(X)$ as a singleton set from the start.)

Computing directly which FDs and MVDs are implied by a set F of FDs and MVDs requires redefining X^+ and $DEP(X)$ to take account of the effects of axioms C1 and C2. For these redefinitions, there exists a polynomial-time algorithm to compute X^+ and $DEP(X)$, from which $F \models X \rightarrow Y$ or $F \models X \twoheadrightarrow Y$ can be decided. In Chapter 8 we shall develop another method to test if an FD or MVD follows from F.

7.5 FOURTH NORMAL FORM

We know that any relation $r(R)$ that satisfies the MVD $X \twoheadrightarrow Y$ decomposes losslessly onto the relation schemes $X\ Y$ and $X\ Z$, where $Z = R - (X\ Y)$. However, if $X \twoheadrightarrow Y$ is the only dependency on R, then R is in 3NF. Therefore, 3NF is not guaranteed to find all possible decompositions.

Definition 7.5 An MVD $X \twoheadrightarrow Y$ is *trivial* if for any relation scheme R with $X\ Y \subseteq R$, any relation $r(R)$ satisfies $X \twoheadrightarrow Y$.

It is left to the reader to show that the trivial MVDs on a relation $r(R)$ are exactly those of the form $X \twoheadrightarrow Y$ where $Y \subseteq X$ or $X\ Y = R$ (see Exercise 7.14). If $X \twoheadrightarrow Y$ is trivial, and we attempt to decompose a relation $r(R)$ using it, one of the projected relations will be all of r. There is no benefit in such a decomposition.

Definition 7.6 An MVD $X \twoheadrightarrow Y$ *applies* to a relation scheme R if $X\,Y \subseteq R$.

Definition 7.7 Let F be a set of FDs and MVDs over $\mathbf{U}$. A relation scheme $R \subseteq \mathbf{U}$ is in *fourth normal form* (4NF) with respect to F if for every MVD $X \twoheadrightarrow Y$ implied by F that applies to R either the MVD is trivial or X is a superkey for R. A database scheme $\mathbf{R}$ is in *fourth normal form* with respect to F if every relation scheme R in $\mathbf{R}$ is in fourth normal form with respect to F.

Example 7.11 Let relation scheme R = FLIGHT DAY-OF-WEEK PLANE-TYPE and let F = {FLIGHT $\twoheadrightarrow$ DAY-OF-WEEK}. R is not in 4NF with respect to F. The data-base scheme $\mathbf{R}$ = FLIGHT DAY-OF-WEEK, FLIGHT PLANE-TYPE is in 4NF with respect to F. Any relation $r(R)$ that satisfies F decomposes losslessly onto the relation schemes in $\mathbf{R}$.

Let us consider the case where we have the MVD $X \twoheadrightarrow Y$ holding on relation scheme R, but X is a key of R. For any relation $r(R)$ the projections

$$r_1 = \pi_{XY}(r) \text{ and } r_2 = \pi_{XZ}(r),$$

where $Z = R - (X\,Y)$, both have the same number of tuples as r. There are no duplicate X-values in r, so there are as many X-values as tuples. Any projection containing the attributes in X must contain all the different X-values.

There is never anything to be gained by such a decomposition. $X\,Y$-values and $X\,Z$-values are not duplicated in r, so no redundancy is removed by the decomposition. No space is saved either. Assuming that each entry in a relation takes one unit of storage space, the relation r takes $|\,r\,| \cdot |\,R\,|$ units (where $|\,r\,|$ is the number of tuples in r). The relations r_1 and r_2 together take $|r| \cdot (|R_1| + |R_2|)$.

Example 7.12 Let $F = \{A \rightarrow B\,C,\ C \twoheadrightarrow D\,E\}$ be a set of dependencies over the relation scheme $R = A\,B\,C\,D\,E$. R is not in 4NF with respect to F because of the MVD $C \twoheadrightarrow D\,E$. The database scheme $\mathbf{R}$ consisting of the two relation schemes

$$R_1 = A\,B\,C \quad \text{and} \quad R_2 = C\,D\,E$$

is in 4NF with respect to F, even though the MVD $A \twoheadrightarrow B$ is implied by F and applies to R_1. $A \twoheadrightarrow B$ is not trivial, but A is a key for R_1.

We can construct 4NF database schemes from a relation scheme R and a set F of FDs and MVDs by decomposition in much the same way we constructed 3NF database schemes. We start with relation R and look for a nontrivial MVD $X \rightarrow\rightarrow Y$ implied by F, where X is not a key for R. We split R into the two relation schemes

$$R_1 = X\,Y \quad \text{and} \quad R_2 = X\,Z,$$

where $Z = R - (X\,Y)$. The MVD $X \rightarrow\rightarrow Y$ is now trivial on R_1 and does not apply to R_2. If either of R_1 or R_2 is not in 4NF with respect to F, we repeat the decomposition process on the offending scheme. Since the MVDs we are using are not trivial, both newly formed relation schemes have fewer attributes than the original relation scheme. Therefore, the decomposition process eventually halts.

Let **R** be a 4NF database scheme obtained by decomposition from a relation scheme R and let F be a set of FDs and MVDs. Any relation $r(R)$ that satisfies F decomposes losslessly onto the relation schemes in **R** (see Exercise 7.15).

Example 7.13 Let $F = \{A \rightarrow\rightarrow B\,C\,D,\ B \rightarrow A\,C,\ C \rightarrow D\}$ be a set of dependencies over the relation scheme $R = A\,B\,C\,D\,E\,I$. Since $A \rightarrow\rightarrow B\,C\,D$ is a nontrivial MVD and A is not a key for R, we decompose R into the relation schemes

$$R_1 = A\,B\,C\,D \quad \text{and} \quad R_2 = A\,E\,I.$$

R_2 is in 4NF with respect to F. F implies the MVD $B \rightarrow\rightarrow A\,C$ on R, but this MVD is not a candidate for use in decomposition because B is a key for R_1, since $C \rightarrow D$. However, $C \rightarrow D$ implies the MVD $C \rightarrow\rightarrow D$, which we can use to decompose R_1. The result is the relation schemes

$$R_{11} = A\,B\,C \quad \text{and} \quad R_{12} = C\,D.$$

Both of these schemes are in 4NF with respect to F. The database scheme $\mathbf{R} = \{R_{11}, R_{12}, R_2\}$ is thus in 4NF with respect to F.

7.6 FOURTH NORMAL FORM AND ENFORCEABILITY OF DEPENDENCIES

We now ask if, for a set of FDs and MVDs F, we can always find a database scheme in 4NF with respect to F upon which F is enforceable. The first problem is that the question is not quite well-posed. The definition of enforceability we use for FDs does not make sense for MVDs.

A set of FDs F is enforceable on a database scheme $\mathbf{R}$ if there is a set of FDs G equivalent to F such that G applies to $\mathbf{R}$. This definition is a reasonable one for FDs for the following reason. Suppose $\mathbf{R}$ is a database scheme over $\mathbf{U}$ and d is a database on $\mathbf{R}$ that is the projection of a single relation $r(\mathbf{U})$. If we can find the actual functional relationship for each FD $X \rightarrow Y$ in G (that is, the corresponding Y-value for each X-value) from d, and $G \models V \rightarrow W$, then we can recover the actual functional relationship for $V \rightarrow W$ from d. The relationship can be reconstructed following the inference axioms as they are applied to derive $V \rightarrow W$ from G (see Exercise 7.16).

The same property is almost true for MVDs. The problem is the complementation axiom, M7. Consider the data base scheme $\mathbf{R} = \{R_1, R_2\}$, where $R_1 = A\ B$ and $R_2 = C$, and the set $F = \{A \twoheadrightarrow B\}$. Suppose d is a database on $\mathbf{R}$ obtained by projecting a relation $r(A\ B\ C)$. We can recover the multivalued relationship for $A \twoheadrightarrow B$ in r from d. However, $F \models A \twoheadrightarrow C$, but we cannot reconstruct the multivalued relationship for $A \twoheadrightarrow C$ from d. Any definition of enforceability for MVDs must deal with the problem of complementation.

Even if we can arrive at an appropriate definition of enforceability for MVDs, we still are not assured of having 4NF and enforceability, as the next result shows. (Recall that in Example 6.26 we saw a set of FDs that was not enforceable on any BCNF scheme.)

Lemma 7.2 If a relation scheme R is in 4NF with respect to a set F of FDs and MVDs, then R is in BCNF with respect to the set of FDs implied by F.

Proof Suppose R is not in BCNF. Then we must have subsets K, Y, and A of R such that K is a key for R, $A \not\subseteq K\ Y$ and $K \rightarrow Y$, $Y \not\rightarrow K$ and $Y \rightarrow A$ under F. The FD $Y \rightarrow A$ implies the MVD $Y \twoheadrightarrow A$. Y is not a key for R, since $Y \not\rightarrow K$. $Y \twoheadrightarrow A$ is not trivial, since A is not contained in Y and $Y A \neq R$, because there must be some attribute B in K that is not in Y. Therefore, R is not in 4NF with respect to F.

There have been attempts at finding a synthetic approach to constructing 4NF database schemes from a set of MVDs and FDs. So far, these attempts have not met with as much success as the synthesis schemes for FDs alone.

7.7 JOIN DEPENDENCIES

MVDs are an attempt to detect lossless decompositions that will work for all relations on a given relation scheme. However, MVDs are not completely adequate in this regard. A relation can have a nontrivial lossless decomposition onto three schemes, but have no such decomposition onto any pair of schemes (see Exercise 6.7). By Theorem 7.1, such a relation satisfies only trivial MVDs (see Exercise 7.17).

Example 7.14 The relation $r(A\ B\ C)$ in Figure 7.1 decomposes losslessly

$r(A$	B	$C)$
a_1	b_1	c_1
a_1	b_2	c_2
a_3	b_3	c_3
a_4	b_3	c_4
a_5	b_5	c_5
a_6	b_6	c_5

Figure 7.1

onto the relation schemes $A\ B$, $A\ C$, and $B\ C$. The projections are shown in Figure 7.2. However, r satisfies no nontrivial MVDs, so it has no lossless

$\pi_{AB}(r) =$

A	B
a_1	b_1
a_1	b_2
a_3	b_3
a_4	b_3
a_5	b_5
a_6	b_6

$\pi_{AC}(r) =$

A	C
a_1	c_1
a_1	c_2
a_3	c_3
a_4	c_4
a_5	c_5
a_6	c_5

$\pi_{BC}(r) =$

B	C
b_1	c_1
b_2	c_2
b_3	c_3
b_3	c_4
b_5	c_5
b_6	c_5

Figure 7.2

decomposition onto any pair of relation schemes R_1 and R_2 such that $R_1 \neq A\ B\ C$ and $R_2 \neq A\ B\ C$.

Definition 7.8 Let $\mathbf{R} = \{R_1, R_2, \ldots, R_p\}$ be a set of relation schemes over $\mathbf{U}$. A relation $r(\mathbf{U})$ satisfies the *join dependency* (JD) $*[R_1, R_2, \ldots, R_p]$ if r decomposes losslessly onto $R_1, R_2, \ldots, R_p$. That is,

$$r = \pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie \cdots \bowtie \pi_{R_p}(r).$$

We also write $*[R_1, R_2, \ldots, R_p]$ as $*[\mathbf{R}]$.

Example 7.15 Relation r in Figure 7.1 satisfies the JD $*[A\ B,\ A\ C,\ B\ C]$.

A necessary condition for a relation $r(\mathbf{U})$ to satisfy the JD $*[R_1, R_2, \ldots, R_p]$ is that $\mathbf{U} = R_1 R_2 \cdots R_p$. We also see from the definition that an MVD is a special case of a JD. A relation $r(R)$ satisfies the MVD $X \twoheadrightarrow Y$ if and only if r decomposes losslessly onto $X\ Y$ and $X\ Z$, where $Z = R - (X\ Y)$. This condition is just the JD $*[X\ Y,\ X\ Z]$. Looking from the other direction, the join dependency $*[R_1, R_2]$ is the same as the MVD $R_1 \cap R_2 \twoheadrightarrow R_1$.

We can also define JDs in a manner similar to the definition of MVDs. Let r satisfy $*[R_1, R_2, \ldots, R_p]$. If r contains tuples $t_1, t_2, \ldots, t_p$ such that

$$t_i(R_i \cap R_j) = t_j(R_i \cap R_j)$$

for all i and j, then r must contain a tuple t such that $t(R_i) = t_i(R_i)$, $1 \leq i \leq p$.

Example 7.16 Suppose relation $r(A\ B\ C\ D\ E)$ satisfies the JD $*[ABC,\ BD,\ CDE]$ and contains the three tuples shown below. Using our

	r(A	B	C	D	E)
t_1	a	b	c	d	e
t_2	a'	b	c'	d'	e''
t_3	a''	b'	c	d'	e'

alternative characterization of JDs, we see that r must also contain the tuple $t = \langle a\ b\ c\ d'\ e' \rangle$.

We shall not present inference axioms for JDs. In Chapter 8 we shall see a method for testing if a set of FDs and JDs (including MVDs) implies a given JD.

7.8 PROJECT-JOIN NORMAL FORM

The point of seeking lossless decomposition is to remove redundancy from relations. We have seen lossless decompositions that do not correspond to MVDs, hence 4NF is not the ultimate in terms of finding lossless decompositions. We shall first define project-join normal form with only decomposition in mind. We then modify the definition slightly to meet another criterion.

Definition 7.9 A JD $*[R_1, R_2, \ldots, R_p]$ over R is *trivial* if it is satisfied by every relation $r(R)$.

We leave it to the reader to show that the trivial JDs over R are JDs of the form $*[R_1, R_2, \ldots, R_p]$ where $R = R_i$ for some i (see Exercise 7.22).

Definition 7.10 A JD $*[R_1, R_2, \ldots, R_p]$ *applies* to a relation scheme R if $R = R_1 R_2 \cdots R_p$.

Definition 7.11 Let R be a relation scheme and let F be a set of FDs and JDs over R. R is in *project-join normal form* (PJNF) with respect to F if for every JD $*[R_1, R_2, \ldots, R_p]$ implied by F that applies to R, the JD is trivial or every R_i is a superkey for R. A database scheme $\mathbf{R}$ is in *project-join normal form* with respect to F if every relation scheme R in $\mathbf{R}$ is in project-join normal form with respect to F.

Example 7.17 Let $F = \{*[A\ B\ C\ D,\ C\ D\ E,\ B\ D\ I],\ *[A\ B,\ B\ C\ D,\ A\ D],\ A \rightarrow B\ C\ D\ E,\ B\ C \rightarrow A\ I\}$ be a set of dependencies over the relation scheme $R = A\ B\ C\ D\ E\ I$. R is not in PJNF with respect to F because of the JD $*[A\ B\ C\ D,\ C\ D\ E,\ B\ D\ I]$. The database scheme $\mathbf{R} = \{R_1, R_2, R_3\}$, where $R_1 = A\ B\ C\ D$, $R_2 = C\ D\ E$, and $R_3 = B\ D\ I$, is in PJNF with respect to F. The JD $*[A\ B,\ B\ C\ D,\ A\ D]$ is implied by F and applies to R_1, but each set of attributes is a superkey for R_1. The MVDs implied by the FDs are either trivial or have keys as left sides.

The reason for allowing a JD $*[R_1, R_2, \ldots, R_p]$ to apply to a relation scheme R and not violate PJNF when every R_i is a superkey is the same as for 4NF. If every R_i is a key, then all projections of a relation $r(R)$ onto the R_i's will have the same number of tuples as r and no redundancy will be removed.

The definition of PJNF above is a weaker condition than the original definition of PJNF as given by Fagin. Besides eliminating redundancy, the original definition ensures enforceability of dependencies by satisfying keys.

Definition 7.12 (revised) Let R be a relation scheme and let F be a set of FDs and JDs. R is in *projection-join normal form* (PJNF) with respect to F if for every JD $*\ [R_1, R_2, \ldots, R_p]$ implied by F that applies to R, $*[R_1, R_2, \ldots, R_p]$ is implied by the key FDs of R.

We leave it to the reader to show that the revised definition is stronger than the first one given (see Exercise 7.24). The following example shows it is strictly stronger.

Example 7.18 Let $R = A\ B\ C$ and let $F = \{A \rightarrow B\ C,\ C \rightarrow A\ B,\ *[A\ B,\ B\ C]\}$. Since $A\ B$ and $B\ C$ are superkeys of R, R satisfies the first definition of PJNF. However, R does not satisfy the revised definition (see Exercise 7.25a).

PJNF implies 4NF, so PJNF and enforceability of dependencies are not always compatible (see Exercise 7.23). PJNF schemes can be constructed by decomposition of a relation scheme using the JDs that cause PJNF violations as guides. We shall see in Chapter 8 how to test when a set of FDs implies a JD.

7.9 EMBEDDED JOIN DEPENDENCIES

Given a relation $r(R)$ and an FD $X \rightarrow Y$, if $X \rightarrow Y$ holds on $\pi_S(r)$, for $X\ Y \subseteq S \subseteq R$, then $X \rightarrow Y$ holds on all of r. The same is not true for JDs, as the next example shows.

Example 7.19 Consider the relation $r(A\ B\ C\ D)$ shown in Figure 7.3. The projection $\pi_{A\ B\ C}(r)$ satisfies the MVD $A \rightarrow\!\!\!\rightarrow B$, but r itself does not.

$r(A$	B	C	$D)$
a	b	c	d
a	b'	c	d
a	b	c'	d'
a	b'	c'	d
a'	b'	c'	d'

Figure 7.3

Definition 7.13 Relation $r(R)$ satisfies the *embedded join dependency* (EJD) $*[R_1, R_2, \ldots, R_p]$ if $\pi_S(r)$ satisfies $*[R_1, R_2, \ldots, R_p]$ as a regular JD, where $S = R_1 R_2 \cdots R_p$. We allow $R = S$. That is, every JD is an EJD. We also write the embedded multivalued dependency (EMVD) $*[X\ Y,\ X\ Z]$ as $X \rightarrow\!\!\!\rightarrow Y\ (Z)$ (read "X multivalued determines Y in the context of Z").

Example 7.20 The relation r in Figure 7.3 satisfies the EMVD $A \rightarrow\!\!\!\rightarrow B\ (C)$.

No complete axiomatizations are known for EJDs although complete proof procedures exist for classes of dependencies containing EJDs.

7.10 EXERCISES

7.1 Modify the relation r below to satisfy the MVDs $A \twoheadrightarrow B\, C$ and $C\, D \twoheadrightarrow B\, E$ by adding rows.

r(A	B	C	D	E)
a	b	c	d	e
a	b'	c	d'	e
a'	b	c	d	e'

7.2 Prove Lemma 7.1.

7.3 Prove that if the relation $r(R)$ satisfies the MVDs $X \twoheadrightarrow Y_1$, $X \twoheadrightarrow Y_2$, $\ldots, X \twoheadrightarrow Y_k$, where $R = X\, Y_1\, Y_2 \cdots Y_k$, then r decomposes losslessly onto the relation schemes $X\, Y_1, X\, Y_2, \ldots, X\, Y_k$.

7.4 Let $r(R)$ be a relation where $R_1 \subseteq R$, $R_2 \subseteq R$ and $R = R_1 R_2$. Prove that $r = \pi_{R_1}(r) \bowtie \pi_{R_2}(r)$ if and only if $c_R[X{=}x](r) = c_{R_1}[X{=}x](r) \cdot c_{R_2}[X{=}x](r)$ for every X-value x in r.

7.5 Prove that if a relation $r(R)$ satisfies $X \twoheadrightarrow Y$ and $Z = R - (X\, Y)$, then

$$\pi_Z(\sigma_{X=x}(r)) = \pi_Z(\sigma_{X\,Y=xy}(r))$$

for every $X\, Y$-value $x\, y$ in r.

7.6 Prove the equivalence of the two definitions of MVDs.

7.7 Characterize the set of MVDs implied by the single FD $X \rightarrow Y$.

7.8 Prove the correctness of inference axioms M1 and M2.

7.9 Let r be a relation on scheme R and let W, X, Y, Z be subsets of R. Show that if r satisfies $X \twoheadrightarrow Y$ and $Z \subseteq W$, then r satisfies $X\, W \twoheadrightarrow Y\, Z$.

7.10 Prove the correctness of inference axiom M6 using axioms M1-M5 and M7.

7.11 Let r be a relation on scheme R and let X, Y, Z be subsets of R. Show that if r satisfies $X \twoheadrightarrow Y$ and $X\, Y \rightarrow Z$, then r satisfies $X \rightarrow Z - Y$.

7.12 Give a set of FDs and MVDs from which an FD can be derived using axiom C2 that cannot be derived using axioms F1-F6.

7.13 Find $DEP(A\, C)$ under the set $F = \{A \twoheadrightarrow E\, I,\ C \twoheadrightarrow A\, B\}$ of MVDs over the relation scheme $R = A\, B\, C\, D\, E\, I$.

7.14 Show that an MVD $X \twoheadrightarrow Y$ over R is trivial if and only if $X \supseteq Y$ or $X\, Y = R$.

7.15 Let **R** be a 4NF database scheme obtained by decomposition from a relation scheme R and a set F of FDs and MVDs. Show that any relation $r(R)$ that satisfies F decomposes losslessly onto the relation schemes in **R**.

7.16 Let $\mathbf{R} = \{R_1, R_2, \ldots, R_p\}$ be a database scheme over **U** and let $d = \{r_1, r_2, \ldots, r_p\}$ be a database over **R** that is the projection of a single relation $r(\mathbf{U})$. That is, $r_i = \pi_{R_i}(r)$, $1 \leq i \leq p$. Show that if the FDs $X \rightarrow Y$ and $Y \rightarrow Z$ apply to **R**, then it is possible to recover the actual functional relationship for $X \rightarrow Z$ in r from d.

7.17 Show that a relation $r(R)$ has no lossless decompositions onto any pair of relation schemes R_1 and R_2, where $R_1 \neq R$ and $R_2 \neq R$, if and only if r satisfies only trivial MVDs.

7.18 Give an example of a relation $r(A\ B\ C\ D\ E)$ that satisfies the JD $*[A\ B\ C,\ B\ D\ E,\ A\ C\ E]$ but satisfies no nontrivial MVD.

7.19 What does it mean for a relation r to satisfy a JD $*[R_1, R_2, \ldots, R_p]$ where all the R_i's are disjoint?

7.20 Let relation r satisfy $*[R_1, R_2, \ldots, R_p]$. If $t_1, t_2, \ldots, t_p$ are tuples in r such that $t_i(R_i \cap R_j) = t_j(R_i \cap R_j)$ for all i and j, show that $t_1', t_2', \ldots, t_p'$ are joinable, where $t_i' = t_i(R_i)$.

7.21 Let $*[R_1, R_2, \ldots, R_p]$ and $*[S_1, S_2, \ldots, S_q]$ be JDs such that for each R_i, $1 \leq i \leq p$, there exists an S_j such that $R_i \supseteq S_j$. Show that $*[S_1, S_2, \ldots, S_q]$ implies $*[R_1, R_2, \ldots, R_p]$.

7.22 Show that a JD $*[R_1, R_2, \ldots, R_p]$ over R is trivial if and only if $R = R_i$ for some i.

7.23 Show that PJNF implies 4NF.

7.24 Show that the revised definition of PJNF implies the first definition given.

7.25 Refer to Example 7.18.
   (a) Give a relation over R with keys A and C that violates $*[A\ B,\ B\ C]$.
   (b) Show that decomposing a relation over R that satisfies F onto $\{A\ B,\ B\ C\}$ requires more space than the original relation.

7.26 Show that the JD $*[A\ B\ C,\ B\ D\ E,\ A\ E\ I]$ over $A\ B\ C\ D\ E\ I$ implies the EJD $*[A\ B,\ B\ E,\ A\ E]$, but not the EJD $*[B\ C,\ B\ D,\ A\ I]$.

7.11 BIBLIOGRAPHY AND COMMENTS

Multivalued dependencies were introduced by Fagin [1977c]. The same concept, under a different name, was independently put forth by Zaniolo [1976] and Delobel [1978]. Beeri, Fagin, and Howard [1977] introduced the first complete axiomatization for FDs and MVDs. Mendelzon [1979] discusses independence of these axioms. The construction of $DEP(X)$ is from Beeri [1980]. Hagihara, Ito, *et al.* [1979], Galil [1979], and Sagiv [1980] give efficient algorithms for calculating *DEP(X)*. Beeri [1979], Biskup [1978, 1980b], and Zaniolo [1979] also discuss inference rules for MVDs, particularly the applicability of complementation in the context of databases. Fischer, Jou, and Tsou [1981] discuss succinct representations for sets of MVDs. Katsuno [1981b] treats some semantic aspects of MVDs.

Fagin [1977a] introduced the fourth normal form. Beeri [1979] and Kambayashi [1979] give synthesis algorithms that incorporate MVDs. Beeri and Vardi [1981a] point out some problems in achieving 4NF. Other normal forms and decompositions strategies are treated by Armstrong and Delobel [1980], Fagin [1980b], Lien [1981], Namibar [1979], Tanaka, Kambayashi, and Yajima [1979a], and Zaniolo and Melkanoff [1981, 1982].

Join dependencies were first introduced in full generality by Rissanen [1977] and were extensively studied by Aho, Beeri, and Ullman [1979]. Sciore [1982] axiomatizes a class of dependencies slightly larger than the class of JDs. Beeri and Vardi [1981b] and Vardi [1980a, 1980b] give inference axioms for FDs and JDs together. Project-join normal form is from Fagin [1979].

Embedded cases of MVDs were recognized by both Fagin [1977c] and Delobel [1978]. Parker and Parsaye-Ghomi [1980], and Sagiv and Walecka [1979] showed there is no complete, finite axiomatization for EMVDs alone. Tanaka, Kambayashi, and Yajima [1979b] also discuss EMVDs. Sadri and Ullman [1980a] give a proof procedure for a more general class of dependencies (which we shall take up in Chapter 14). Beeri and Vardi [1980a, 1980b] and Chandra, Lewis, and Makowsky [1981] showed that implication for this class of dependencies is undecidable. There is no contradiction here, since the proof procedure is for finite and infinite relations. An implication statement can have an infinite counterexample but no finite counterexample.