THE SEMANTIC JUSTIFICATION FOR NORMAL FORMS IN RELATIONAL DATABASE DESIGN

By

Millist Walter Vincent

B.Sc.(Hons.), Dip.Comp.Sc., M.E.

Department of Computer Science

Monash University

Thesis Submitted for Examination

for the Degree of

Doctor of Philosophy

1994

TABLE OF CONTENTS

CHAPTER 1

THESIS OVERVIEW 1

1.1. Background and Motivation 1

1.2. Previous Work On Justifying the Use of Normal Forms 3

1.3. Contribution of the Thesis 7

1.4. Scope of the Research 8

1.5. Structure of the Thesis 9

1.5.1. Redundancy 9

1.5.2. Key-Based Update Anomalies 11

1.5.3. Unpredictable Insertions 15

1.5.4. Fact-Based Update Anomalies 16

1.6. Numbering System 19

CHAPTER 2

THE RELATIONAL MODEL 20

2.1. Introduction 20

2.2. Data Definition 20

2.3. Data Operators 21

2.4. Relational Constraints 22

2.4.1. Projected and Embedded Dependencies 25

2.4.2. Closure and Dependency Basis 26

2.4.3. Pure MVDs and Reduced Covers 28

2.4.4. Keys 30

2.4.5. Join Dependencies 31

2.5. Tableau 32

2.6. The History and Definitions of Normal Forms 35

2.6.1. Third Normal Form (3NF) 36
2.6.2. Boyce-Codd Normal Form (BCNF) 38
2.6.3. Fourth Normal Form (4NF) 39
2.6.4. (3,3)NF 42
2.6.5. Projection-Join Normal Form (PJNF) 44
2.6.6. Elementary Key Normal Form (EKNF) 45
2.6.7. Domain-Key Normal Form (DK/NF) 47
2.7. Desirable Database Designs 49

CHAPTER 3

REDUNDANCY AND NORMAL FORMS 53
3.1. Introduction 53
3.2. The Definition of Redundancy 55
3.3. Certain Properties of Normal Forms 58
3.4. The Case of FD Constraints 64
3.5. The Case of MVD Constraints 67
3.6. The Case of FD and MVD Constraints 69
3.7. Related Work 75
3.8. Conclusions 78

CHAPTER 4

KEY-BASED UPDATE ANOMALIES AND NORMAL FORMS 80
4.1. Introduction 80
4.2. The Definitions of Key-Based Update Anomalies 85
4.2.1. Insertion Anomaly 85
4.2.2. Deletion Anomaly 86
4.2.3. Modification Anomalies 88
4.3. The Case of FD Constraints 91

4.3.1. Insertion Anomaly and Normal Forms 92
4.3.2. MA_1 Anomaly and Normal Forms 94
4.3.3. MA_2 Anomaly and Normal Forms 97
4.3.4. MA_3 Anomaly and Normal Forms 102
4.3.4. Comparing PANF With Other Normal Forms 104
4.3.6. Achieving PANF 105
4.4. The Case of MVD Constraints 108
4.4.1. Insertion Anomaly and Normal Forms 108
4.4.2. Deletion Anomaly and Normal Forms 109
4.5. The Case of FD and MVD Constraints 111
4.5.1. Insertion Anomaly and Normal Forms 111
4.5.2. Deletion Anomaly and Normal Forms 112
4.5.3. MA_1 and MA_2 Anomalies and Normal Forms 116
4.5.4. MA_3 Anomaly and Normal Forms 123
4.6. Related Work and Conclusions 130

CHAPTER 5

UNPREDICTABLE INSERTIONS AND NORMAL FORMS 134
5.1. Introduction 134
5.2. The Definitions of Unpredictable Insertions 137
5.3. The Case of MVD Constraints 142
5.3.1. The Relationship between Insertion Anomalies 142
5.3.2. Insertion Normal Forms and 4NF 145
5.4. The Case of FD and MVD Constraints 147
5.4.1. The Relationship between Insertion Anomalies 147
5.4.2. Insertion Anomalies and 4NF 148
5.5. Conclusions 152

CHAPTER 6

FACT-BASED UPDATE ANOMALIES AND NORMAL FORMS 155

6.1. Introduction 155

6.2. The Definitions of Update Anomalies and Fact-Based Normal Forms........... 161

6.2.1. Modification Anomalies 161

6.2.2. Replacement Anomalies 167

6.3. The Case of FD Constraints 170

6.3.1. Modification Anomalies and Normal Forms 170

6.3.2. Replacement Anomalies and Normal Forms 174

6.4. The Case of MVD Constraints 177

6.4.1. Modification Anomalies and Normal Forms 177

6.4.2. Replacement Anomalies and Normal Forms 181

6.5. The Case of FD and MVD Constraints.. 183

6.5.1. Modification Anomalies and Normal Forms 183

6.5.2. Replacement Anomalies and Normal Forms 186

6.6. Related Work and Discussion 187

6.7. Conclusions 189

CHAPTER 7

CONCLUSIONS AND FUTURE WORK... 191

7.1. Conclusions 191

7.2. Future Work.. 194

7.2.1. Normal Forms and Join Dependencies 194

7.2.2. Normal Forms and Null Values 195

7.2.3. Multiple Relations 196

7.2.4. Normal Forms in other Data Models.. 197

REFERENCES 200

SUMMARY

Relational database systems, as a result of their simplicity and sound theoretical basis, have become widely used in industry in recent years. Associated with their use is the issue of how to correctly structure or design the data to be used in a relational database. Depending on the type of constraints or rules which apply to data items in the database, several criteria, referred to as *normal forms*, have been proposed as conditions that a database design should satisfy to ensure an absence of processing difficulties with the database.

Although the definitions of the most widely used normal forms have existed for some time, the issues of formally defining the processing difficulties that normal forms are intended to solve, as well as proving that the normal forms are exactly the conditions needed to avoid these difficulties, have not been completely resolved. A few researchers have addressed this question, but most of them have focused on the simplest case where the only constraints on a relation are *functional dependencies* (FDs). The contribution of this thesis is to place the use of normal forms on a firmer theoretical foundation, firstly by completing previous work on justifying the use of normal forms for the case of FD constraints and then by extending the same approach to the case where the constraints can include a more general type of constraint, called a *multivalued dependency* (MVD).

Motivations for four different types of desirable properties that a relation scheme should possess are presented, their properties are formally defined and their relationship to normal forms is investigated. The properties proposed are: an *absence of redundancy*, an absence of *key-based update anomalies*, an absence of *unpredictable insertions* and an absence of *fact-based update anomalies*. The motivation for these properties are: an absence of redundancy ensures that a relation will not contain duplicate information; an absence of key-based update anomalies ensures that as long as a simple type of

constraint, called a key constraint, is satisfied by a relation after an update then the new relation will automatically satisfy all the more complex types of constraints (such as FD and MVDs); an absence of unpredictable updates ensures that in making insertions to a relation, different insertions do not require different information to be supplied; and lastly, an absence of fact-based update anomalies ensures that the atomic units of information stored in a relation can be independently updated. For each of these types of desirable properties, several subtypes, corresponding to different choices for the set of constraints, are also defined .

The main results derived in the thesis are that for some desirable processing properties, the traditional normal forms of *Boyce-Codd normal form* (BCNF) and *fourth normal form* (4NF) are both necessary and sufficient conditions to guarantee these properties. However, for other desirable processing properties, BCNF and 4NF are shown to be stronger than what is required and the necessary and sufficient conditions are new normal forms which are different from any of the normal forms so far defined in the literature.

DECLARATION

I declare that the thesis contains no material which has been accepted for the award of any degree or diploma in any university and that, to the best of my knowledge, the thesis contains no material previously published or written by another person except where due reference is made in the text.

Signed ....

Date ....

Department of Computer Science,
Monash University,
Clayton, Victoria, 3168.
1994

ACKNOWLEDGMENTS

I would firstly like to thank my supervisor, Professor Bala Srinivasan, for his support, encouragement and careful reading of the thesis. I am also grateful to the Department of Computer Science at Monash University for allowing me the flexible arrangements that have made this thesis possible. Thanks are also due to Professor Chris Marlin at Flinders University where part of the work was carried out while on study leave. I am also indebted to Dr. Chris Steketee and my colleagues in the School of Computer and Information Science at the University of South Australia for many discussions and helpful suggestions. Finally, I would like to dedicate this work to my wife Therese and my daughters Anna and Elise.

SYMBOL INDEX

Symbol	Meaning
V, W, X, Y, Z	sets of attributes names
$A, B, C, \ldots$	attribute names
DOM(A)	the domain of an attribute A
R	the set of attribute names in a relation
$r(R)$	a relation defined over R
t	a tuple in a relation
$t[X]$	the projection of a tuple t onto a set of attributes X
$\pi_X[r]$	the projection of a relation r onto a set of attributes X
$r_1 \bowtie r_2$	the natural join of two relations r_1 and r_2
$X \rightarrow Y$	X functionally determines Y
$X \rightarrow\rightarrow Y$	X multidetermines Y
Σ	a set of functional and multivalued dependencies
Σ^+	the set of dependencies logically implied by Σ
Σ'	$\Sigma' = \Sigma \cup \{X \rightarrow\rightarrow R - XY \mid X \rightarrow\rightarrow Y \in \Sigma\}$
Σ_k	the set of key dependencies
SAT(Σ)	the set of all relations which satisfy Σ
ATT(d)	the set of attributes appearing in the functional or multivalued dependency denoted by d
$\equiv$	logical equivalence
X^+	the closure of the set of attributes X
DEP(X)	the dependency basis of X
$*[R_1, R_2, \ldots, R_p]$	a join dependency
T	a tableau
V_d	the set of distinguished variables
V_n	the set of nondistinguished variables

$chase_{\Sigma}(T)$	the tableau resulting from applying the chase algorithm to a tableau T
$\neg$	not

CHAPTER 1

THESIS OVERVIEW

1.1. BACKGROUND AND MOTIVATION

Since the original definition of the relational data model by Codd [Codd 1970], relational databases have since become widely used in commercial applications. A central issue that arises with the use of relational database systems is that of how to design or structure the information that the database is to contain. Broadly speaking, this problem of relational database design can be stated as follows: starting with a set of constraints which represent the business rules governing the relationships between data items of the application, how does one split the application data into separate relations in such a way that the relations have desirable processing properties?

This problem was first addressed by Codd for the case where the only type of constraint is a *functional dependency* (*FD*) [Codd 1972] and he proposed a precise set of criteria - called *first normal form* (*1NF*), *second normal form* (*2NF*) and *third normal form* (*3NF*) - which he proposed a relation scheme should satisfy in order to avoid certain processing difficulties and data redundancies occurring in relations defined over the scheme. These normal forms, which will be referred to in this thesis as *syntactic normal forms*, are expressed as conditions on the set of attributes that appears in a relation scheme and the set of constraints that apply to the scheme. Since Codd's original definitions, many other normal forms have been proposed that either improve Codd's original definitions for the case of FD constraints - such as *Boyce-Codd normal form (BCNF)* [Codd 1974], *(3,3)NF* [Smith 1978], *elementary key normal form (EKNF)* [Zaniolo 1982] or *improved third normal form* [Ling et al. 1981] - or extend them to more general classes of constraints, such as *fourth normal form (4NF)* in the case of *multivalued dependencies* (*MVDs*) [Fagin 1977c], *projection-join normal form* (*PJNF*) [Fagin 1979] in the case of *join dependencies* (*JDs*) [Rissanen 1979] and *domain key normal form (DK/NF)* [Fagin 1981] in the case of constraints expressible in first-order logic.

The motivation for the work in this thesis grew out of the observation (also noted by others [Biskup 1989; Thalheim 1988; Vossen 1990]) that in spite of the fact that much research has been devoted to normalisation and relational database design, the issue of providing a formal justification for the use of normal forms, especially for the case where the constraints include MVDs, was far from complete. In most of the works on normal forms just mentioned, the approach taken has been to provide a precise definition of a normal form but to justify it by way of examples rather than by formally deriving the normal form condition from a precisely defined processing property.

The aim of this thesis is to place the use of normal forms in relational databases on a much sounder theoretical foundation by adopting a different and more rigorous approach than the one just mentioned. In our approach, we firstly address the issues of determining what the desirable processing properties of a relation are and analysing why these properties are desirable, then we formally define these properties (which will be referred to as *semantic normal forms*). As will be seen later, while these semantic normal forms encapsulate the desirable processing properties of a relation scheme, because of the way in which they are formulated they are not useful for checking if a relation scheme has these properties since in general they require an infinite number of relations to be tested. Hence the other part of our approach is to formally derive from these semantic normal forms equivalent, but more practically useful, syntactic normal forms and compare them with the other syntactic normal forms which have been defined in the literature.

The main results derived in this thesis, which will be discussed in more detail in later sections of this and other chapters, are that for some semantic normal forms, the traditional syntactic normal forms of BCNF and 4NF are equivalent to them, i.e. the traditional syntactic normal forms are both necessary and sufficient conditions on a relation scheme to ensure desirable processing properties in all relations defined over the scheme. However, for other semantic normal forms, the traditional normal forms are stronger than what is required and the equivalent syntactic normal forms are shown to be different from any syntactic normal forms that have been so far defined in the literature. Finally, we note that much of the work contained in the thesis has also been reported in the literature [Vincent 1991; Vincent 1992a; Vincent 1992b; Vincent and Srinivasan 1992a; Vincent and Srinivasan 1992b; Vincent and Srinivasan 1993a; Vincent and Srinivasan 1993b; Vincent and Srinivasan 1993d; Vincent and Srinivasan 1994a; Vincent and Srinivasan 1994b; Vincent and Srinivasan 1994c].

1.2. PREVIOUS WORK ON JUSTIFYING THE USE OF NORMAL FORMS

The pioneering work on normalisation and database design was done by Codd [Codd 1972]. In his approach, a tuple is not regarded as the atomic unit of information; instead, the value of a tuple on the set of attributes in a constraint (which we shall refer to as a *fact*) is regarded as the atomic unit of information for retrieval and update. Based on this interpretation, Codd considered three types of processing difficulties that can occur in a relation, which he referred to as *insertion dependencies*, *deletion dependencies* and *update dependencies*[1]. Essentially, all these anomalies occur when facts cannot be independently manipulated by the corresponding database update operations without violating the properties of the relational model or the set of constraints. Codd then defined 3NF and justified it by an example which showed that the processing anomalies

---

[1]Since Codd's work, the terminology generally used in the literature, which shall also be adopted here, has changed. What Codd called a *dependency* is now referred to as an *anomaly* and the term *update anomaly* is used to collectively describe all the different types of anomalies and not the specific anomaly which occurs when the contents of a tuple are changed (this anomaly will be referred to as a replacement anomaly).

he identified could be avoided if the relation scheme was split into 3NF schemes. However, Codd's approach was essentially intuitive and he neither formally defined update anomalies, nor proved that either 3NF is a sufficient condition for a relation scheme to avoid processing anomalies in relations defined over the scheme or is a necessary condition to avoid anomalies. In a later paper [Codd 1974], Codd defined a new normal form, called BCNF, which is a stronger condition than 3NF, and conjectured, but didn't prove, that BCNF was a necessary and sufficient condition for update anomalies to be absent in a relation scheme.

The first formalisation of the concept of an update anomaly was given by Bernstein and Goodman [Bernstein and Goodman 1980], although their interpretation of update anomalies differed from that given by Codd. In their paper, precise definitions of the three types of update anomalies (insertion anomalies, deletion anomalies and replacement anomalies) were given and it was proven that BCNF is a necessary and sufficient condition on a relation scheme for the avoidance of each of the types of update anomaly. They also considered the usefulness of BCNF in the context of multiple relations and showed that in this setting, having individual relation schemes in BCNF does not guarantee an absence of processing difficulties. However, these conclusions should be treated with caution since central to their results and conclusions was the restrictive assumption that all relations are the projection of a single universal relation, called the *universal instance assumption* (*UIA*), which is now regarded as being incorrect. Later, it was shown [Jajodia and Ng 1983] that if one replaced the UIA assumption by the less restrictive and now widely accepted *weak instance approach* [Honeyman 1980; Honeyman 1982; Sagiv 1981], then most of the problems encountered by Bernstein and Goodman in the context of multiple relations disappear.

More recently, Chan, Vossen and Biskup have all considered the relationship between update anomalies and normal forms. In Chan's work [Chan 1989], formal definitions of update anomalies were given which were intended to formalise the original fact-based approach by Codd mentioned earlier, but was slightly more general than Codd's approach since Chan allowed for the possibility that facts may not correspond to FDs. He then showed that if the facts do corresponded to FDs, then BCNF is a necessary and sufficient condition for a relation scheme to have no insertion or deletion anomalies. In the case of a replacement anomaly, he showed that an absence of this anomaly is an equivalent condition to BCNF, or single key BCNF, depending on which attributes are assumed to be allowed to be modified. Based on the weak instance approach mentioned earlier, Chan also investigated necessary and sufficient conditions for an absence of a replacement anomaly in the context of multiple relations. He showed that in this context, stronger conditions than the requirement that the individual relation schemes be in BCNF are required to ensure an absence of replacement anomaly because of the possibility of inter-relation dependencies causing difficulties even when the individual schemes are in BCNF.

Vossen [Vossen 1988] also gave definitions of update anomalies and showed that BCNF is an equivalent condition to the absence of an update anomaly in the relation scheme. However, nothing essentially new was added in this work since the definitions and results are equivalent to those of Bernstein and Goodman mentioned earlier, except that they were derived in a different fashion.

Biskup [Biskup 1989] presented another approach to justifying the use of BCNF. He also considered sets of attributes to be the fundamental units of information (which he referred to as *objects*), but regarded these as being the sets of attributes in the left-hand sides of FDs rather than all the attributes in an FD. He then proposed that objects in a relation should satisfy two conditions, namely a uniqueness condition that requires that object values in a relation be unique and either a strong independence condition which requires that every tuple with a unique value for an object can always be inserted into a relation without violating the FD constraints, or a weak independence condition that requires that for any distinct value of an object, there exists some tuple with the same object value which can be inserted into the relation without violating the FD constraints. Based on these conditions, he defined two object normal forms and showed that one of the normal forms is equivalent to BCNF and the other is equivalent to single key BCNF. In a later paper [Biskup and Dublish 1991], this approach was extended to the multiple relation setting with inclusion dependencies also being allowed [Casanova et al. 1984; Mitchell 1983a; Mitchell 1983b; Sciore 1983b] in order to model set inclusion constraints.

A different approach to justifying the use of normal forms, based on *keys* and enforcing *key uniqueness*, is due to Fagin [Fagin 1979; Fagin 1981]. In this interpretation, a relation scheme is defined to have an update anomaly if an update to a relation defined over the scheme results in candidate key uniqueness being maintained (no two tuples in the relation having the same value for any candidate key) but some general constraint, such as an FD or MVD, being violated. His rationale for this approach is that key uniqueness can be, and is, relatively easily enforced by most commercial relational systems, but checking that general constraints are satisfied is much more computationally difficult and such a facility is not available in relational software. Therefore, he argued that it is desirable that the satisfaction of all constraints on a relation be a logical consequence of key uniqueness, since then the consistency of the database after an update can be ensured by enforcing key uniqueness alone. Conversely, a key-based update anomaly is considered undesirable because its occurrence implies that the consistency of the database cannot be guaranteed by only enforcing key uniqueness. Further support for this approach is given by his results [Fagin 1979] that the BCNF, 4NF and PJNF conditions on a relation scheme all have the property that the relevant set of constraints (FDs for BCNF, FDs and MVDs for 4NF, JDs for PJNF) are automatically satisfied by any relation defined over the scheme if the key uniqueness condition is satisfied. These results give a justification to BCNF, 4NF and PJNF since they imply that if a relation scheme is in one of these normal forms then no key-based insertion anomaly can occur with respect to a set of constraints of the relevant type. In his later paper [Fagin 1981], Fagin extended this approach still further and considered another type of constraint, called a *domain constraint* (which is a condition that an attribute value lie in a specific set) along with key-uniqueness to be what he termed *elementary constraints*. He then defined a relation scheme to be in the normal form DK/NF if, for every relation defined over the scheme, all the constraints on the relation, which Fagin also allowed to include arbitrary sentences in first-order logic, are a consequence of the elementary constraints.

1.3. CONTRIBUTION OF THE THESIS

The thesis contains a systematic investigation of the semantic justification for the syntactic normal forms BCNF and 4NF which extends the previous research on this topic reported in Section 1.2. In particular, much attention is devoted to the justification of 4NF since this topic has been the focus of relatively little research. In particular, the thesis contains the following contributions:

- We formally define *redundancy* in a relation scheme and investigate the derivation of equivalent syntactic normal forms which ensure its absence. The redundancy property is a straightforward formalisation of the property often given informally in introductory texts for justifying the use of normal forms. It also uses the fact-based interpretation of the semantics of the data in a relation discussed in the previous section and in more detail later in Section 1.5 and Chapter 3.

- The relationship between normal forms and the key-based update anomalies defined by Fagin is extended to include a new type of key-based update anomaly called a *modification anomaly*. As will be discussed in more detail later in Section 1.5 and Chapter 4, a *key-based modification anomaly* occurs when the key values are preserved during an update to a relation and key uniqueness is maintained but the constraints are still violated. The relationship between an absence of this new type of key-based update anomaly and the normal forms BCNF and 4NF is analysed and it is shown that for certain classes of modification anomalies, necessary and sufficient condition for their absence are new syntactic normal forms which have not appeared before in the literature. New results which extend those of Fagin concerning the relationship between two particular types of key-based update anomalies - insertion and deletion anomalies - and normal forms are also derived for the situation where the constraints contain both FDs and MVDs.

- The approach of Bernstein and Goodman to justifying normal forms, discussed in the previous section, is extended to the cases where the set of constraints includes only MVDs or both FDs and MVDs.

- New definitions of *fact-based replacement anomalies* which improve previous definitions are proposed. The relationship between their absence and BCNF and 4NF is then examined.

1.4. SCOPE OF THE RESEARCH

In this investigation of the justification for normalisation, the scope of the work has the following restrictions:

- The only types of constraints allowed are FDs and MVDs.
- Only single relations and relation schemes are considered.
- Relations do not include null values.
- The size of attribute domains is assumed to be infinite. This assumption avoids some undesirable interactions between constraints and finite domains [Atzeni and DeAntonellis 1993].

1.5. STRUCTURE OF THE THESIS

In this section, the structure of the thesis will be summarised. The next chapter (Chapter 2) contains background material on the relational model and a review of all the syntactic normal forms that have so far been defined in the literature. The technical content of the thesis is contained in Chapters 3 - 6 and the final chapter, Chapter 7, contains some concluding remarks and directions for further research. An outline of the contents of Chapters 3 - 6 is now presented.

1.5.1. Redundancy

In Chapter 3, we investigate the problem of data redundancy in relations and relation schemes and we derive syntactic normal forms which guarantee its absence. The redundancy property is a straightforward formalisation of the informal justification for normalisation often given in database texts [Date 1990; Ullman 1988a] yet, somewhat surprisingly, has not previously been formally related to normalisation. The approach we take in defining redundancy in this chapter is based on viewing a constraint, such as an FD or MVD, as not only representing enterprise rules governing the way different data items in a database are related, but also as representing the fundamental unit of information (fact) for retrieving and updating the data in a relation. For example, given the relation scheme $\{COURSE, CNAME, TEACHER, TEXT\}$ with the meaning that a tuple $<a, b, c, d>$ over the scheme represents the information that a course with code a and course name b is taught by a teacher c and uses a text book d. If the set of dependencies which apply to the scheme is $\{COURSE \rightarrow CNAME, COURSE \rightarrow\rightarrow TEACHER\}$ where $\rightarrow$ denotes an FD and $\rightarrow\rightarrow$ denotes an MVD, then one possible set of facts is $\{\{COURSE, CNAME\}, \{COURSE, TEACHER\}\}$. This interpretation of the semantics of the data in a relation is a natural generalisation of the approach originally due to Codd [Codd 1972] and, as will be discussed in more detail in Chapter 3, has since been widely used in several areas of database theory. A relation scheme is then defined to be *redundant* if there exists a *legal relation* (satisfies the set of constraints) defined over the scheme which contains at least two tuples which are identical on a fact. For instance, in the example just given, the relation scheme is redundant since the relation shown in Figure 1.1 satisfies the set of constraints but has two or more tuples which are identical on the fact $\{COURSE, TEACHER\}$.

COURSE	CNAME	TEACHER	TEXT
1	Physics	Green	Optics
1	Physics	Allan	Mechanics
1	Physics	Green	Mechanics
1	Physics	Allan	Optics
2	Maths	Green	Mechanics
2	Maths	Green	Calculus

Figure 1.1. A relation containing redundancy

However, a subtle point that arises in this approach to defining redundancy is that there are several possible choices for the set of facts. As was done in the above example, one could simply choose the set of facts to be the sets of attributes in the FD and MVD constraints derived from the database design. However, it is well known [Fagin 1977c] that MVDs have the symmetrical property that an MVD $X \rightarrow\rightarrow Y$ is satisfied in a relation if and only if the MVD $X \rightarrow\rightarrow R - XY$ is also satisfied, thus there is no real basis for preferring XY as a fact to $XR - XY$ and so both could be considered to be facts. In the present example, this interpretation would result in the set of facts being $\{\{COURSE, CNAME\}, \{COURSE, TEACHER\}, \{COURSE, TEXT, CNAME\}\}$. The last possibility is to extend the set of facts still further and to allow any dependencies which are logically implied by the original set of constraints to be facts. For instance, the set of constraints in the current example also implies dependencies such as *COURSE*

$TEACHER \rightarrow CNAME$, $COURSE \rightarrow\rightarrow TEXT$ and many others. We won't write out the set of all implied dependencies in full because, even in this simple example, the number of implied dependencies is large. Since there is no real basis for any one of these sets of facts being more correct than the others, we allow for each of them and correspondingly define three different types of redundancy in a relation scheme. As mentioned in Section 1.1, these redundancy properties, as with other desirable properties to be discussed later, although formally encapsulating what is meant by redundancy in a relation, are not directly useful for testing a relation scheme since they result in all relations defined over the scheme (which is infinite if the sets of attribute values are assumed to be infinite) having to be tested. The main content of the chapter is then to derive syntactic normal forms which are equivalent conditions to the absences of the three types of redundancy in a relation scheme and to compare these syntactic normal forms to BCNF and 4NF.

1.5.2. Key-Based Update Anomalies

In Chapter 4, the relationship between normal forms and the absence of key-based update anomalies in a relation scheme is analysed. This key-based view of normalisation is based on the work of Fagin [Fagin 1979; Fagin 1981]. As mentioned earlier, the motivation for this approach is that the enforcement of key uniqueness can be performed efficiently and such a facility is available in most relational software, whereas a facility for enforcing the satisfaction of arbitrary constraints (such as FDs or MVDs) is not, and thus a desirable property of a relation is that the satisfaction of the general constraints be implied by the satisfaction of the key constraints. Conversely, a key-based update anomaly is considered to occur if some update[2] to a relation results in key uniqueness

[2]Update is here used in a generic sense and means either the insertion, deletion or modification of a tuple in a relation.

being satisfied but a general constraint being violated. The following example illustrates a key-based update anomaly in the case of the update being an insertion.

Example 1.1. Let the relation scheme $R = \{EMP, DEPT, MNGR\}$ and let the set of constraints be $\{EMP \rightarrow DEPT, DEPT \rightarrow MNGR\}$. The only candidate key is EMP and the relation r shown in Figure 1.2 satisfies the set of constraints. However R has a key-based insertion anomaly since the insertion of a tuple t with the value <*Cauchy, Math, Euler*> into the relation r results in the new relation, r', satisfying the key uniqueness property but violating the FD $DEPT \rightarrow MGR$. □

r

EMP	DEPT	MNGR
Hilbert	Math	Gauss
Laplace	Math	Gauss
Turing	Physics	Bohr

insert <Cauchy, Math, Euler>

⇓

r'

EMP	DEPT	MNGR
Hilbert	Math	Gauss
Laplace	Math	Gauss
Turing	Physics	Bohr
Cauchy	**Math**	**Euler**

Figure 1.2. An example of an insertion anomaly

In this chapter we propose a new type of key-based update anomaly called a modification anomaly. We define an update to a relation to be a *modification violation* if the modification of a tuple in the relation results in the violation of the general constraints although both key uniqueness and a new condition - that the *identity* of the tuple be preserved by the modification - are satisfied, and then a relation scheme is defined to have a *modification anomaly* if there exists a modification violation to a legal relation defined over the scheme. The additional condition, that the identity of the tuple be preserved, is motivated by the observation that in practice it is often undesirable to change the identity of a tuple because of the need to also update associated foreign key references as well as possible confusion as to which real world entity the tuple refers to. In the relational model, a candidate key has the property of being a unique identifier and so it is natural to equate the identity of a tuple with its value on a candidate key. In general, however, a relation scheme may have several candidate keys and so there are several possibilities as to what could be interpreted as the identity of a tuple. The three possibilities considered are: (i) at least one (arbitrary) candidate key of the original tuple is unchanged by the modification; (ii) the primary key of the original tuple is unchanged by the modification; (iii) all candidate keys of the original tuple are unchanged by the modification. According to each of these possibilities, three different types of modification anomaly are defined. These concepts are now illustrated by the following example.

Example 1.2. Consider the case where the relation scheme is $\{A, B, C, D\}$ and the set of constraints is $\{ABC \rightarrow D, D \rightarrow C, B \rightarrow\rightarrow A\}$. It can be verified that the candidate keys are ABC and ABD. The relation r shown in Figure 1.3 satisfies the set of constraints.

If the tuple $t = \langle a_2, b_1, c_1, d_1 \rangle$ is changed to $t^* = \langle a_2, b_1, c_1, \boldsymbol{d_2} \rangle$, resulting in the relation r' shown in Figure 1.3, then r has a modification violation and hence the relation scheme has a modification anomaly of the first type mentioned above (one candidate key is unchanged by the modification). This is because the relation r satisfies the set of constraints but the new relation, r', is unique on the keys ABC and ABD, t and t^* are identical on the key ABC but r' violates the constraint $B \rightarrow\rightarrow A$. □

r

A	B	C	D
a_1	b_1	c_1	d_1
a_2	b_1	c_1	d_1

replace $<a_2, b_1, c_1, d_1>$ by $<a_2, b_1, c_1, \mathbf{d}_2>$

$\Downarrow$

r'

A	B	C	D
a_1	b_1	c_1	d_1
a_2	b_1	c_1	$\mathbf{d}_2$

Figure 1.3. An example of a modification anomaly

The main contribution of the chapter is to derive, for each type of modification anomaly, syntactic normal forms which are equivalent to the absence of the modification anomaly for two classes of constraints. The first class is the case where the only dependencies are FDs and the second is where the set of dependencies contains both FDs and MVDs. The case where the only constraints are MVDs is not considered since no modification anomaly can occur for this case. As will be discussed in more detail in Chapter 4, two of the syntactic normal forms derived are not equivalent to any of the syntactic normal forms which have been defined in the literature. The other contributions of the chapter are to strengthen a result of Fagin [Fagin 1981] concerning normal forms and insertion anomalies as well as providing a new result relating 4NF and the absence of a deletion anomaly in a relation scheme.

1.5.3. Unpredictable Insertions

In Chapter 5, the justification for normalisation is viewed from the perspective of avoiding unpredictable insertions to a relation. As mentioned in Section 1.2, this approach was originally formulated by Bernstein and Goodman [Bernstein and Goodman 1980] and also later by Vossen [Vossen 1988] in a slightly different fashion. The motivation for this approach is now outlined. Consider the relation scheme $R = \{EMP, DEPT, MNGR\}$ used earlier in Example 1.1 and the following relation defined over it.

EMP	DEPT	MNGR
Hilbert	Math	Gauss
Laplace	Math	Gauss
Fermi	Physics	Bohr

Figure 1.4. An example of a relation having an unpredictable insertion

If one wants to insert the information that the employee *Laplace* works in the *Math* department then no value has to be supplied for manager since it is already known that *Gauss* is the manager of the Math department. However, if one wants to add the fact *Turing* works in the *Computing* department then a new value has to be added for the manager since the manager of the *Computing* department cannot be deduced from the current information in the relation. So the relation scheme in this example is said to have an *unpredictable insertion* because two different insertions to a relation defined over the scheme require different information to be supplied. To be more precise and using the terminology of Bernstein and Goodman, the insertion of a tuple t into a relation r *affects* a set of attributes X if the projection of r onto X is not equal to the projection of $r \cup \{t\}$ onto X, and conversely X is *unaffected* by the insertion if the projections are the same. A relation scheme is then defined to have an *unpredictable insertion* if there exists at least two different insertions on relations defined over the scheme such that the attributes in an

FD or MVD constraint are affected by one of the insertions but not by the other. We note that, as in the case of redundancy, this approach is based on interpreting the set of attributes in a constraint as being the fundamental unit of information. We also allow for the same three possible choices for the set of constraints as we use in our investigation of redundancy. These are: the set Σ of FDs and MVDs derived from the database design, Σ plus all MVDs $X \rightarrow\rightarrow R - XY$ corresponding to the MVDs $X \rightarrow\rightarrow Y$ in Σ; and lastly, Σ^+, the set of all FDs and MVDs logically implied by Σ. Corresponding to each of these choices, three semantic normal forms are defined on relation schemes such that a scheme is in one of the normal forms if no relation defined over it has an unpredictable insertion with respect to the relevant set of facts. In the case where the only constraints are FDs, the results of Bernstein and Goodman and Vossen show that the three semantic normal forms are all equivalent conditions on a relation scheme and are equivalent to BCNF. In Chapter 5 we investigate the problem of deriving syntactic normal forms which are equivalent to the semantic ones just mentioned for the cases where the set of constraints contains only MVDs, and when it contains both FDs and MVDs.

1.5.4. Fact-Based Update Anomalies

In Chapter 6, the relationship between normal forms and several types of what we call a fact-based update anomaly is analysed. This approach to justifying normalisation is closest to the original intuitive justification of normal forms proposed by Codd. As in our approach to redundancy and unpredictable updates discussed previously, this approach is based on interpreting the set of attributes in an MVD or FD constraint as the fundamental unit of information for update. In essence, a fact-based update anomaly occurs when a relation stores the values of several independent facts with the result that these values cannot be updated independently without violating either the basic properties of the relational model or a general constraint on the relation.

As mentioned in Section 1.2, Chan [Chan 1989] provided formal definitions of the three types of fact-based update anomalies - *insertion anomaly, deletion anomaly and* *replacement anomaly* - and investigated their relationship to normal forms for the case of FD constraints. The main contribution of the chapter is to propose definitions for two different types of a replacement anomaly, ones which we feel are more consistent with the basics of the relational model than the one given by Chan, and then investigate the relationship between their absence in a relation scheme and the syntactic normal forms BCNF and 4NF. The first type of replacement anomaly, simply called a replacement anomaly, is said to occur in a relation scheme when the replacement of the value of a fact in a tuple of a legal relation defined over the scheme results in key-uniqueness being maintained but an FD or MVD dependency being violated. The following example illustrates this definition.

Example 1.3. Consider the relation scheme $R = \{ A, B, C \}$with the set of constraints being $\{A \to B, B \to C\}$. A legal relation, r, defined over R is shown in Figure 1.5. Then R has a replacement anomaly since when the value of BC in the tuple <*2, 2, 1*> is changed to <*2,* ***1, 2***>, the resulting relation, r', satisfies the key uniqueness condition since both tuples in r' are different on the only candidate key A, but r' violates the FD $B \to C$. □

r

A	B	C
1	1	1
2	2	1

replace <2, 2, 1> by <2, **1**,**2**>

⇓

r'

A	B	C
1	1	1
2	**1**	**2**

Figure 1.5. An example of a replacement anomaly

The other contribution of Chapter 6 is to define several subtypes of a more restrictive type of replacement anomaly and investigate the relationship between their absence in a relation scheme and the normal forms BCNF and 4NF. Motivated by the works of Biskup and the *entity-relationship approach* to data modelling [Batini et al. 1991; Biskup 1989; Biskup and Dublish 1991; Chen 1976] where instead of regarding the set of attributes in a dependency as an indivisible unit of information, the approach adopted is to view the sets of attributes X and Y in a dependency $X \rightarrow Y$ or $X \rightarrow\rightarrow Y$ as playing different roles. X is interpreted as representing some entity and the attributes in Y are interpreted as the properties of X. We then consider fact replacements where the Y-values can change, but not the X-values which represent the identity of an entity in this approach. A modification anomaly is then defined to be a replacement anomaly where the only attribute values which can be modified are those belonging to the right-hand side of a dependency.

1.6. NUMBERING SYSTEM

The thesis is divided into seven chapters. Each chapter is divided into subsections and these are sometimes divided into further subsections. A section labelled as 3.2.1 is thus the first subsection of the second subsection of Chapter 3. The labelling system for examples, definitions, figures and results is to prefix them by the chapter number and then to label sequentially within the chapter. Thus Definition 5.11 refers to the 11th definition in Chapter 5 and Lemma 4.8 refers to the 8th lemma presented in Chapter 4.

CHAPTER 2

THE RELATIONAL MODEL

2.1. INTRODUCTION

In this chapter we shall outline the basic relational concepts and results that will be used in later chapters. Thorough presentations can be found in standard database texts such as those by Ullman or Maier [Maier 1983; Ullman 1988a].

The relational model was defined first by Codd as an abstract model and since then, as a result of its simplicity, flexibility and rigorous foundations, it now forms the basis of most commercially used database systems [Codd 1970]. The relational model consists of three parts: data definition, data operators and integrity constraints. We now present each of these parts in turn.

2.2. DATA DEFINITION

A universe U is a fixed, finite set of symbols, called attributes, which represent the column names in a relation. As usual, the symbols $A, B, C, \ldots$ and their subscripts represent single attributes and $V, W, X, \ldots$ and their subscripts denote sets of attributes. The union of the attribute sets X and Y is denoted by XY rather than $X \cup Y$. $X - Y$ denotes set difference. The *domain* of an attribute $A \in U$, denoted by DOM(A), is a set of values representing the possible values that can appear in an A column. We assume that for any attribute A, DOM(A) is infinite. This assumption is not always crucial to our results, though we note that certain difficulties can arise if the domains are too small

[Atzeni and DeAntonellis 1993; Fagin 1981; Ginsburg and Zaiddan 1982; Kanellakis 1980].

A *relation scheme* R is a subset of U. Let the elements of a relation scheme R be the set denoted by $\{A_1, \ldots, A_n\}$. A *tuple* over R is an element of $\mathrm{DOM}(A_1) \times \ldots \times \mathrm{DOM}(A_n)$ where $\times$ denotes the cartesian product. A *relation instance* (or simply a *relation*) over R, denoted by $r(R)$, is a *finite* set of tuples defined over R. In this paper, all relations are defined over a single relation scheme R and so $r(R)$ will often be denoted simply by r. These definitions are illustrated in the following example.

Example 2.1. Let $U = \{SCODE, SNAME, TEACHER, TEXT\}$, $\mathrm{DOM}(SCODE)$ = Integer, $\mathrm{DOM}(SNAME)$= Char(10), $\mathrm{DOM}(TEACHER)$ = Char(10), $\mathrm{DOM}(TEXT)$ = Char(10) and $R = \{SCODE, SNAME, TEACHER, TEXT\}$. A relation defined over R is illustrated in Figure 2.1. A tuple $<a, b, c, d>$ in the relation represents the information that a subject with code a and name b is taught by a teacher c and uses a text book d. □

SCODE	SNAME	TEACHER	TEXT
1	Physics	Green	Optics
1	Physics	Allan	Mechanics
1	Physics	Green	Mechanics
1	Physics	Allan	Optics
2	Maths	Green	Mechanics
2	Maths	Green	Calculus

Figure 2.1. An example of a relation

2.3. DATA OPERATORS

Codd [Codd 1970] defined a set of operators for data manipulation and later [Codd 1972] demonstrated the power of the this set by proving that it is equivalent in expressive power to a subset of first order logic (called safe queries). For the purpose of this work, the full set of relational operators is not needed and so we shall only define those operators that are required.

If t is a tuple over R and X is a subset of R, then $t[X]$ is the *restriction* of t to the attributes in X. If r is a relation over R, then the *projection* of r onto X, denoted by $\pi_X[r]$, is the relation defined by:

$$\pi_X[r] = \{t[X] \mid t \in r\}$$

Let $r_1(R_1)$ and $r_2(R_2)$ be two relations. The *natural join* of r_1 and r_2, denoted by $r_1 \bowtie r_2$, is a relation defined over R_1R_2 satisfying the condition:

$$r_1 \bowtie r_2 = \{t(R_1R_2) \mid \exists\, t_1, t_2\ ((t_1 \in \mathrm{r}_1) \text{ and } (\mathrm{t}[R_1] = t_1) \text{ and } (t_2 \in \mathrm{r}_2) \text{ and }$$
$$(\mathrm{t}[R_2] = t_2) \text{ and } (t_1[R_1 \cap R_2] = t_2[R_1 \cap R_2]))$$

2.4. RELATIONAL CONSTRAINTS

In this thesis, we consider only two types of constraints - functional dependencies and multivalued dependencies. While much more general classes of constraints have been defined [Beeri and Vardi 1984a; Beeri and Vardi 1984b], FDs and MVDs are the most important types of dependencies in practical database design [Delobel and Adiba 1985; Thalheim 1991] and it is for this reason that we restrict ourselves to these classes. Extensive surveys of dependency theory and discussions of more general types of dependencies than the ones examined in this study are given in the work by Fagin and Vardi as well as that by Thalheim [Fagin and Vardi 1986; Kanellakis 1990; Thalheim 1991].

A *functional dependency* (*FD*) constraint, originally introduced by Codd [Codd 1972], is a constraint denoted by $X \rightarrow Y$ where X and Y are sets of attributes. A relation r *satisfies* the FD $X \rightarrow Y$ if for all tuples $t_1, t_2 \in r$, if $t_1[X] = t_2[X]$ then $t_1[Y] = t_2[Y]$; otherwise it *violates* the FD. For example, the relation shown in Figure 2.1 satisfies the FD $SCODE \rightarrow SNAME$ but violates the FD $SCODE \rightarrow TEACHER$.

A *multivalued dependency* (*MVD*), originally introduced independently by Fagin and Zaniolo [Fagin 1977c; Zaniolo 1976], is a constraint denoted by $X \rightarrow\rightarrow Y$. A relation r satisfies the MVD $X \rightarrow\rightarrow Y$ if for all $t_1, t_2 \in r$ with $t_1[X] = t_2[X]$, there exists a tuple $t_3 \in r$ such that $t_3[X] = t_1[X]$, $t_3[Y] = t_1[Y]$ and $t_3[R - XY] = t_2[R - XY]$. For example, the relation in Figure 2.1 satisfies the MVDs $SCODE \rightarrow\rightarrow TEACHER$ and $SCODE \rightarrow\rightarrow TEXT$. We shall assume that X and Y in any MVD $X \rightarrow\rightarrow Y$ are disjoint because of the result [Fagin 1977c] that $X \rightarrow\rightarrow Y$ is satisfied if and only if $X \rightarrow\rightarrow Y - X$ is satisfied. When we want to explicitly represent all the individual attributes in either an FD or MVD, we shall do it by using the abbreviated notation such as, for example, $SCODE\ TEACHER \rightarrow\rightarrow TEACHER\ TEXT$ rather than the full set notation $\{SCODE, TEACHER\} \rightarrow\rightarrow \{TEACHER, TEXT\}$. The set of all relations which satisfy Σ, a set of FDs and MVDs, is denoted by SAT(Σ). The set of attributes which are in either the left-hand side or right-hand side of a dependency $d \in \Sigma$ is denoted by ATT(d). A dependency d applies to a relation scheme R if ATT(d) $\subseteq R$. A set Σ of FDs and MVDs *apply* to a relation scheme R if every dependency in Σ applies to R. Since we are only dealing with a single relation scheme in this work, we will assume that a set of FDs and MVDs always apply to the relation scheme in question.

A dependency is *trivial* in a relation scheme R if it is satisfied by every relation defined over R. It can be shown [Maier 1983] that an FD $X \rightarrow Y$ is trivial if and only if $Y \subseteq X$, and an MVD $X \rightarrow\rightarrow Y$ is trivial if and only if $Y \subseteq X$ or $R = XY$.

Even though we will not pursue this issue in this work since our focus is on the use of dependencies in database design, we note that it is also possible to view FDs and MVDs in a much more abstract setting and several researchers have investigated this issue [Demetrovics et al. 1992; Lakshmanan and VeniMadhavan 1987; Lee 1983; Matus 1991; Novotny and Novotny 1992].

We now discuss the crucial concepts of implication and derivation with respect to a set of dependencies. Given a set Σ of FDs and MVDs and an FD $Z \rightarrow W$ (or MVD $Z \rightarrow\rightarrow W$), Σ *implies* the FD $Z \rightarrow W$ (or MVD $Z \rightarrow\rightarrow W$) if every relation that satisfies Σ also satisfies $Z \rightarrow W$ (or $Z \rightarrow\rightarrow W$). The set of all FDs and MVDs that are implied by a set Σ of FDs and MVDs is denoted by Σ^+. It is also known [Armstrong 1974; Beeri et al. 1977] that valid dependency implications can be obtained by using proofs involving a finite sequence of *inference rules (axioms)*. The following set has been shown to be a valid set of inference rules for FD and MVD constraints [Beeri et al. 1977].

FD rules:

A1: If $Y \subseteq X$ *then* $X \rightarrow Y$

A2: If $X \rightarrow Z$ *and* $Y \subseteq U$ *then* $XY \rightarrow ZU$

A3: If $X \rightarrow Y$ *and* $Y \rightarrow Z$ *then* $X \rightarrow Z$

MVD rules:

A4: If $X \rightarrow\rightarrow Y$ *then* $X \rightarrow\rightarrow R - XY$

A5: If $X \rightarrow\rightarrow Y$ *and* $V \subseteq W$ *then* $WX \rightarrow\rightarrow VY$

A6: If $X \rightarrow\rightarrow Y$ *and* $Y \rightarrow\rightarrow Z$ *then* $X \rightarrow\rightarrow Z - Y$

A7: If $Y \subseteq X$ *then* $X \rightarrow\rightarrow Y$

Combined FD and MVD rules:

A8: If $X \rightarrow Y$ *then* $X \rightarrow\rightarrow Y$

A9: If $X \rightarrow\rightarrow Y$, $Z \subseteq Y$, $W \cap Y = \emptyset$ *and* $W \rightarrow Z$, *then* $X \rightarrow Z$

The following rules, although derivable from those above, are useful and will be needed in later chapters.

A10: If $X \rightarrow YZ$ *then* $X \rightarrow Y$

A11: If $X \rightarrow\rightarrow Y$ *and* $X \rightarrow\rightarrow Z$ *then* $X \rightarrow\rightarrow YZ$

A critical question that arises with the use of a set of inference rules is to determine whether the set is sufficiently powerful for any implied dependency to be derived by a finite application of a set of inference rules. If the set of inference rules has this property, then it is said to be *complete*. For the case of FD constraints, Armstrong [Armstrong 1974] and Fagin [Fagin 1977b] proved that a set of inference rules which is equivalent to A1-A3 is complete; and Beeri *et al.* [Beeri et al. 1977] proved that rules A1-A9 are complete for FDs and MVDs. However, it should be noted that for some other types of relational constraints, either no complete set of inference rules is known or it has been shown that no complete, finite set of inference rules exists [Chandra and Vardi 1985; Parker and Parsaye-Ghomi 1980; Petrov 1989; Sagiv and Walecka 1982; Sciore 1982].

Another related aspect of dependency implication is to derive, if possible, an algorithm for determining whether a set of dependencies implies another dependency. If such an algorithm exists for a specific class of dependencies, then the implication is said to be *decidable* for that class of dependencies. Although the notion of completeness and decidability are related and many of the algorithms for implication are based on the properties of an axiom system, in general the notions are not equivalent. A more thorough discussion of these concepts is contained in the book by Paredaens *et al.* [Paredaens et al. 1989].

2.4.1. Projected and Embedded Dependencies

In the design of relational databases, the following question often arises. Given a relation scheme and a set Σ of dependencies which apply to it, which dependencies are implied in a subset R' of R? These dependencies are called *projected dependencies* and are formally defined as follows. A dependency d is *implied* in R' if for every relation $r(R) \in \mathrm{SAT}(\Sigma)$, $\pi_{R'}[r]$ satisfies d.

For an FD constraint, the answer of which FDs are implied in R' is quite easy [Maier 1983]. An FD $X \rightarrow Y$ is implied in R' if and only if $XY \subseteq R'$ and $X \rightarrow Y \in \Sigma^+$. For the case of MVDs, the situation is not as simple because, unlike FDs, the validity of an

MVD $X \rightarrow\rightarrow Y$ depends on the attributes in R - XY. For an MVD, it has been shown that an MVD $X \rightarrow\rightarrow Y$ is implied in R' if and only if $XY \subseteq R'$ and there exists Y' such that $Y = R' \cap Y'$ and $X \rightarrow\rightarrow Y' \in \Sigma^+$ [Aho et al. 1979a; Beeri and Vardi 1981b].

We note that if one poses the related, but different, question of whether a relation $\pi_{R'}[r]$ satisfying a dependency d implies that r also satisfies d, then the situation is again more complicated for MVDs. For the case of FDs, the answer is in the affirmative [Maier 1983], but for MVDs it was first noted by Fagin and Deolobel that an MVD can hold in the projection without it holding in the original relation [Delobel 1978; Fagin 1977c; Maier 1983]. Such MVDs are called *embedded MVDs* and have been investigated by several authors [Ito et al. 1983; Parker and Parsaye-Ghomi 1980; Sagiv and Walecka 1982; Tanaka et al. 1979].

2.4.2. Closure and Dependency Basis

The *closure* of a set Σ of FDs and MVDs, denoted by Σ^+, is the set of all FDs and MVDs implied by Σ. For example, if $R = \{A, B, C\}$ and $\Sigma = \{A \rightarrow B\}$ then $\Sigma^+ = \{A \rightarrow A, A \rightarrow B, B \rightarrow B, C \rightarrow C, AB \rightarrow A, AB \rightarrow B, BC \rightarrow B, BC \rightarrow C, AC \rightarrow A, AC \rightarrow C\}$. Since inference rules A1 - A9 are complete, the closure of a set of FDs and MVDs is also equal to the set of FDs and MVDs which can be derived from Σ and inference rules A1-A9. In general, the number of dependencies in the closure is exponentially proportional to the number of dependencies in the original set [Maier 1983]. Two sets of dependencies, Σ and Ψ, are defined to be *equivalent*, written as $\Sigma \equiv \Psi$, if $\Sigma^+ = \Psi^+$. For example [Maier 1983], the set of FDs $\{A \rightarrow BC, A \rightarrow D, CD \rightarrow E\}$ is equivalent to $\{A \rightarrow BCE, A \rightarrow ABD, CD \rightarrow E\}$. If $\Sigma \equiv \Psi$, then Ψ is a *cover* for Σ.

We now introduce the related concepts of the closure and the dependency basis of a set of attributes. The *closure* of a set of attributes X, denoted by X^+, is the set of attributes such that an attribute $A \in X^+$ if $X \rightarrow A$ in Σ^+. It then follows from the inference rules for FDs (rules A1-A3) that an FD $X \rightarrow Y \in \Sigma^+$ if and only if Y is a union of attributes in X^+ [Ullman 1988a]. The *dependency basis* for a set of attributes X, denoted by DEP(X), is a disjoint set of attribute sets such that for every set Y with $X \rightarrow\rightarrow Y \in \Sigma^+$, Y is a union of sets from DEP(X) and there is no other set with a smaller number of attribute sets having the same property. Writing the elements in DEP(X) as $\{X_1, \ldots, X_p, X_1^+, \ldots, X_j^+, W_1, \ldots, W_n\}$, it can be shown [Beeri 1980; Fagin 1977c; Paredaens et al. 1989] that DEP(X) has the following properties:

(i) DEP(X) covers R, i.e. $R = \cup Z_i$ where $Z_i \in$ DEP(X);

(ii) The sets in DEP(X) are disjoint;

(iii) $X \rightarrow\rightarrow Y \in \Sigma^+$ if and only if $Y = \cup Z_i$ where $Z_i \in$ DEP(X);

(iv) $X_1, \ldots, X_p$ are single attribute sets such that $X = \bigcup_{i=1}^{i=p} X_i$;

(v) $X_1^+, \ldots, X_j^+$ are single attribute sets such that $X^+ - X = \bigcup_{i=1}^{i=j} X_i^+$.

One important difference between the structure of the sets X^+ and DEP(X) is that the elements in X^+ consist of single attributes, whereas in general the elements in DEP(X) consist of multiple attributes. This difference is because the inference rules imply that any FD $X \rightarrow YZ$ is equivalent to the set $\{X \rightarrow Y, X \rightarrow Z\}$ and so the right-hand side of any FD can be split into single attributes, whereas this is not the case for MVDs. The problem of developing efficient algorithms for the generation of X^+ and DEP(X) has been investigated by several researchers and several methods have been devised [Beeri 1980; Diederich and Milton 1988; Galil 1982; Hagihara et al. 1979; Ito et al. 1984; Lakshmanan and VeniMadhavan 1985; Lakshmanan and VeniMadhavan 1987; Maier et al. 1981; Parker and Delobel 1979; Sagiv 1980; Vardi 1983]. The following example illustrates the concepts of the closure and the dependency basis of a set of attributes.

Example 2.2. Let $R = \{A, B, C, D, E, F, G\}$ and $\Sigma = \{AB \rightarrow\rightarrow DE, E \rightarrow\rightarrow F, E \rightarrow C\}$. Any of the algorithms in the works just cited can be used to show that if we let $X = AB$, then $X^+ = \{A, B, C\}$ and DEP(X) = $\{A, B, C, DE, F, G\}$. □

2.4.3. Pure MVDs and Reduced Covers

We now introduce the concepts of pure MVDs and reduced sets of FDs and MVDs. The motivation for both these concepts is to reduce a set of dependencies to another equivalent set which has no superfluous information. Firstly we present the definition of a pure set of FDs and MVDs which is due to Jajoida [Jajodia 1986].

Definition 2.1. Let Σ be a set of FDs and MVDs. An MVD $X \rightarrow\rightarrow Y \in \Sigma$ is *pure* if it is nontrivial and neither $X \rightarrow Y$ nor $X \rightarrow R - XY$ is in Σ^+. Σ is *pure* if every MVD in Σ is pure.

The following example illustrates the definition.

Example 2.3. Let $R = \{A, B, C\}$ and $\Sigma = \{A \rightarrow\rightarrow B, B \rightarrow A, B \rightarrow C\}$. It follows then from rules A4 and A9 that $A \rightarrow C \in \Sigma^+$ and so $A \rightarrow\rightarrow B$ is not pure. □

The condition in this definition is a natural one. If an MVD $X \rightarrow\rightarrow Y$ is not pure, then it follows from the inference rules that an equivalent set of dependencies can be obtained by replacing $X \rightarrow\rightarrow Y$ in the set of dependencies by the FD $X \rightarrow R - XY$ or $X \rightarrow Y$. So in that sense, $X \rightarrow\rightarrow Y$ is not a 'true' MVD. A related definition aimed at factoring out MVDs which cannot be derived from FDs appears in the concept of an *envelope set* due to Yuan and Ozsoyoglu [Yuan and Ozsoyoglu 1987; Yuan and Ozsoyoglu 1992b] in their work on desirable 4NF decompositions.

It is also clear that any set of FDs or MVDs has a pure cover. To verify this, it follows by a simple application of the inference rules that if an MVD $X \rightarrow\rightarrow Y$ that is not pure is replaced by either by $X \rightarrow Y$ if $X \rightarrow Y \in \Sigma^+$, or by $X \rightarrow R - XY$ if $X \rightarrow R - XY \in \Sigma^+$, then an equivalent set of dependencies is obtained, and so by repeating the procedure a pure cover can be obtained for any set of FDs and MVDs.

The second concept required is that of a *reduced set* of FDs and MVDs, a concept due to Ozsoyoglu and Yuan [Ozsoyoglu and Yuan 1987b] which extends the concept of a minimal cover for a set of FDs [Maier 1983; Ullman 1988a]. Similar concepts were also used by Zaniolo in his work on formalising the database design process [Zaniolo 1976; Zaniolo and Melankoff 1981; Zaniolo and Melankoff 1982]. We now formally define a reduced set of dependencies. We note that our definition is weaker than that of Ozsoyoglu and Yuan since their definition contains the additional condition that no set of attributes be able to be transferred from the left-hand side to the right-hand side of a dependency.

Definition 2.2. Let Σ be a set of FDs and MVDs. Σ is reduced if:

(i) No dependency $d \in \Sigma$ is redundant, i.e. for all $d \in \Sigma$, Σ - $\{d\}$ is not equivalent to Σ ;

(ii) Every dependency is left-reduced, i.e. for every MVD $X \rightarrow\rightarrow Y$ (or FD $X \rightarrow Y$) $\in \Sigma$, there is no MVD $X' \rightarrow\rightarrow Y$ (or FD $X' \rightarrow\rightarrow Y$) $\in \Sigma^+$ such that $X' \subset X$;

(iii) every dependency is right-reduced, i.e. for every MVD $X \rightarrow\rightarrow Y$ (or FD $X \rightarrow Y$) $\in \Sigma$, there is no MVD $X \rightarrow\rightarrow Y'$ (or FD $X \rightarrow\rightarrow Y'$) $\in \Sigma^+$ such that $Y' \subset Y$.

We note that it is easy to establish that a reduced cover can be generated for any set Σ of FDs and MVDs by using the following method. Firstly, any redundant dependencies are removed from Σ. Then, any MVD $X \rightarrow\rightarrow Y$ (or FD $X \rightarrow Y$) in Σ that is not left-reduced is replaced by $X' \rightarrow\rightarrow Y$ (or $X \rightarrow Y$) and any MVD $X \rightarrow\rightarrow Y$ (or FD $X \rightarrow Y$) that is not right-reduced is replaced by the MVDs $X \rightarrow\rightarrow Y'$ (or FD $X \rightarrow Y'$) and $X \rightarrow\rightarrow Y - Y'$ (or FD $X \rightarrow Y - Y'$). It can be easily verified that equivalence is maintained by these replacements. These steps are repeatedly applied until no more changes can be made to the set of dependencies. We also note that it follows easily from the inference rules that if $X \to YZ \in \Sigma$ then $X \to Y \in \Sigma^+$ and so the FDs in a reduced set of FDs and MVDs contain a single attribute on the right-hand side.

2.4.4. Keys

Since there is no notion of an ordering on the tuples in the relational model, keys then play a central role in the retrieval of information because they provide the only method by which a tuple may be identified. We now present definitions for some essential key-related concepts.

Given a relation scheme R and a set Σ of FDs and MVDs which apply to it, a set of attributes X is a *superkey* for a relation scheme R if the FD $X \to R \in \Sigma^+$. X is a *candidate key* if it is a superkey and it has no proper subset X' such that $X' \to R$ is also in Σ^+. An attribute is a *prime attribute* if it is a member of a candidate key. The *key constraints*, denoted by Σ_K, is the set of all FDs in Σ^+ of the form $K \to R$ where K is a candidate key. The set of all relations which satisfy Σ_K is denoted by SAT(Σ_K). Obviously, if a relation satisfies Σ then it also satisfies Σ_K but the converse is not true. Also, it is easily seen that a relation satisfies Σ_K if and only if no tuples in the relation have the same value for a candidate key.

Several researchers have addressed issues concerning algorithms for the generation and testing of key-related properties [Beeri and Bernstein 1979; Demetrovics 1978; Demetrovics and Thi 1988; Forsyth and Fadous 1975; Kambayashi 1979; Lucchesi and Osborn 1978; Pichat 1985; Selesnjew and Thalheim 1988; Thalheim 1992; Thuan 1987; Thuan and Bao 1985; Yu and Johnson 1976]. While generating a single candidate key for a relation scheme can be done in polynomial time [Lucchesi and Osborn 1978], generating all candidate keys for a relation scheme is inherently exponential [Lucchesi and Osborn 1978]. The reason for this is that the number of candidate keys in a relational scheme can be exponentially proportional to the number of attributes and the number of dependencies. As a result, the determination of many key-related properties turns out to be computationally intractable. For example, it has been shown [Lucchesi and Osborn 1978] that problems such as testing whether an attribute is prime or determining if there is a candidate key less than a fixed size are NP-complete [Garey and Johnson 1979].

2.4.5. Join Dependencies

Join dependencies, a generalisation of multivalued dependencies, were first proposed by Rissanen [Rissanen 1979] and are defined as follows. Let $R_1, R_2, \ldots, R_p$ denote sets of attributes of a relation scheme R such that $R = R_1 R_2 \ldots R_p$. A *join dependency* (JD) is a constraint denoted by $*[R_1, R_2, \ldots, R_p]$. A relation r (R) satisfies the join dependency if $r = \pi_{R_1}(r) \bowtie \pi_{R_2}(r) \ldots \bowtie \pi_{R_p}(r)$.

We now briefly discuss the issues of decidability and the existence of a complete axiom system for the implication problem for JDs. The decidability question was settled in the affirmative by Aho *et al.* [Aho et al. 1979a], although from a computational complexity perspective the result was not very encouraging since the algorithm was later shown to be NP-hard [Maier et al. 1981]. In contrast, the completeness question was answered in the negative by a result due to Petrov who proved that there is no finite, complete set of inference axioms for JDs [Petrov 1989]. However, complete sets of inference axioms are known for restricted classes of JDs as well as for dependencies that are more general than JDs [Delobel 1978; Sciore 1982]. Although no complete set of axioms is possible for JD inference, sound sets of axioms have been derived [Beeri and Vardi 1981b; Beeri and Vardi 1985; Sciore 1982].

JDs will not be considered directly in this paper, but the result [Fagin 1977c] that any MVD $X \rightarrow\rightarrow Y$ is equivalent to the JD $*[XY, XZ]$, where $Z = R - XY$, will be used frequently.

2.5. TABLEAU

The last relational concepts required are those of a *tableau* and the *chase algorithm*. The concept of a tableau is originally due to Aho *et al.* where it was used as a means of representing projection-join mappings[Aho et al. 1979a; Aho et al. 1979b; Aho et al. 1979c]. Maier *et al.* [Maier et al. 1979] then showed that the tableau concept can be used in conjunction with an algorithm, called the chase, to provide another method of determining when a set of FDs and JDs implies another JD or FD. This is a more powerful method than using derivations based on inference rules since it can be used to test if a set of FDs and JDs implies a JD, a problem which, as mentioned earlier, cannot be solved by applying a set of inference rules. We will briefly outline the chase algorithm and some results relating to it that will be used later. A more thorough presentation of the these concepts is contained in the text by Maier [Maier 1983].

A *tableau* is a matrix consisting of sets of rows. Each column in the tableau corresponds to an attribute in R. Each row consists of variables drawn from a set V, that is the disjoint union of two sets V_d and V_n. V_d is the set of *distinguished variables* and V_n is the set of *nondistinguished variables*. Any variable is restricted to appear in at most one column and in each column there can be one and only one distinguished variable.

A *valuation* is a function ρ which maps each variable in a tableau T to an element in DOM(A) where A is the column in which the variable appears. This is extended to a function from a tableau T to a relation over R as follows. If $\omega = \langle \nu_1, \nu_2, \ldots \nu_n \rangle$ is a row of T, then $\rho(\omega)$ is the tuple $\langle \rho(\nu_1), \rho(\nu_2), \ldots \rho(\nu_n) \rangle$ and $\rho(T) = \{\rho(\omega) \mid \omega \text{ is a row in } T\}$.

Let Σ be a set of FDs and JDs (any MVD is treated as a JD by the result mentioned earlier). The *chase* is a result of applying the following transformations to a tableau T until no further changes can be made:

F-Rule: For every FD $X \rightarrow A$ in Σ, there is an associated F-rule that transforms the tableau as follows. Suppose that the tableau T has rows ω_1 and ω_2 where $\omega_1[X] = \omega_2[X]$. Let $\nu_1 = \omega_1[A]$ and $\nu_2 = \omega_2[A]$. If either of ν_1 or ν_2 is a distinguished variable and the other is not, then the nondistinguished variable is changed to the distinguished variable. If both are nondistinguished variables, then the one with the larger subscript is replaced by the one with the smaller subscript.

J-Rule: Let $*[R_1, R_2, \ldots, R_p]$ be a JD in Σ. If there exists a row ω such that $\omega[R_1] \in T[R_1], \ldots, \omega[R_p] \in T[R_p]$, ω is added to T.

For an MVD, we use the result quoted that any MVD $X \rightarrow\rightarrow Y$ is equivalent to the JD $*[XY, XZ]$ where $Z = R - XY$. We now illustrate these concepts by the following example .

Example 2.4. Let $R = \{A, B, C, D, E\}$, let $\Sigma = \{B \rightarrow C, C \rightarrow\rightarrow AB\}$ and consider the tableau T shown in Figure 2.2. Distinguished variables are indicated in the tableau by variables with a and nondistinguished variables are indicated by variables with b. The F-rule for $B \rightarrow C$ can be applied to rows 1 and 2 in T to yield the tableau T_1. Then rewriting the MVD $C \rightarrow\rightarrow AB$ as the JD $*[CAB, CDE]$ and applying it to rows 1 and 2 in T_1 yields the tableau T_2. Applying the JD again to rows 1 and 2 in tableau T_2 yields tableau T_3 upon which the chase terminates since no more changes can be made. □

T

A	B	C	D	E
b_1	a_1	b_2	a_2	a_3
a_3	a_1	a_4	b_3	b_4

T_1

A	B	C	D	E
b_1	a_1	a_4	a_2	a_3
a_3	a_1	a_4	b_3	b_4

T_2

A	B	C	D	E
b_1	a_1	a_4	a_2	a_3
a_3	a_1	a_4	b_3	b_4
b_1	a_1	a_4	b_3	b_4

T_3

A	B	C	D	E
b_1	a_1	a_4	a_2	a_3
a_3	a_1	a_4	b_3	b_4
b_1	a_1	a_4	b_3	b_4
a_3	a_1	a_4	a_2	a_3

Figure 2.2. An example illustrating the chase algorithm

Let $chase_\Sigma(T)$ be the tableau which results from applying any F-rules and J-rules that are applicable until no more changes can be made to the tableau. Then it can be shown

[Maier 1983; Maier et al. 1979] that the chase always terminates, is independent of the sequence in which the rules are applied and is unique up to a renaming of nondistinguished variables. The following results [Maier 1983; Maier et al. 1979] concerning properties of the chase will be used in later chapters.

Lemma 2.1. *Any valuation ρ of $chase_{\Sigma}(T)$ which is a one-to-one mapping satisfies Σ.*

Lemma 2.2. *Let T_X be the tableau constructed as follows. It contains two rows, one row, denoted by ω_d, contains all distinguished variables and the other, denoted by ω_X, contains distinguished variables in the X-columns and nondistinguished variables elsewhere. Let $T^* = chase_{\Sigma}(T_X)$ and let ω_d^* and ω_X^* be the rows in T^* that correspond to ω_d and ω_X in T (ω_d^* and ω_X^* may be the same row). Then $\omega_d^* = \omega_d$ and an FD $X \rightarrow Y$ is in Σ^+ iff the Y-columns in T^* contain only distinguished variables.*

Lemma 2.3. *Let $X \rightarrow\rightarrow Y$ be an MVD defined on a relation scheme R, let $Z = R - XY$ and define the tableau $T_{\boldsymbol{R}}$ as follows. $T_{\boldsymbol{R}}$ consists of two rows. One contains distinguished variables in the XY columns and nondistinguished variables elsewhere, while the other contains distinguished variables in the XZ columns and nondistinguished variables elsewhere. Let $T^* = chase_{\Sigma}(T_{\boldsymbol{R}})$. Then $X \rightarrow\rightarrow Y$ is in Σ^+ iff T^* contains a row with distinguished variables only.*

2.6. THE HISTORY AND DEFINITIONS OF NORMAL FORMS

In this section we review the normal forms that have been defined in the literature. The sequence in which they are presented here essentially corresponds to the chronological order in which they appeared in the literature.

2.6.1. Third Normal Form (3NF)

The first formal investigation of file design was done by Codd although some of the difficulties that could result from arbitrary file designs were recognised earlier by Heath [Codd 1972; Heath 1971]. Codd defined both *second normal form* (*2NF*) and 3NF in this work after having proposed *first normal form* (*1NF*) in his initial work on the foundations of the relational model [Codd 1970]. Although we will define 3NF later in a different fashion to that proposed by Codd, it is instructive to quote the original definitions directly from Codd's seminal work:

" A relation R is in *second normal form* if it is in first normal form and every non-prime attribute of R is fully dependent on each candidate key of R";

"Suppose that A, B, C are three distinct collections of attributes of a relation R (hence R is of degree 3 or more). Suppose that all three of the following time-independent conditions hold: R.A $\rightarrow$ R.B, R.B $\not\rightarrow$ R.A, R.B $\rightarrow$ R.C . . . in such a case we shall say that C is *transitively dependent* on A under R";

" A relation R is in *third normal form* if it is in second normal form and every non-prime attribute is non-transitively dependent on each candidate key of R."

It is interesting to note that this definition of a transitive dependency, and thus that of 3NF, differs from what is the accepted definition today. The difference is that although the sets A, B, C in Codd's definition are required to be distinct, there is no requirement that the sets be disjoint and so it is possible, for instance, for one of the sets to be a subset of another. Not excluding this possibility appears to have been intentional on the part of Codd since in a later section he uses the example of a relation scheme $R(A, B, C)$ with the FDs $AB \rightarrow C$, $C \rightarrow B$ and states that B is transitively dependent on the candidate key AB (and so is not in 3NF according to his definition). However, the currently accepted definition of 3NF adds an extra condition to Codd's definition of a transitive dependency - that C not be a subset of A or B. This difference in definition is not purely academic since one of the main methods of generating 3NF relations, the synthesis method [Bernstein 1976], produces 3NF schemes only if the newer definition of 3NF is adopted.

The definition of 3NF that we present here is due to Zaniolo [Zaniolo 1982] who also proved that it is equivalent to Codd's original definition of 3NF provided that Codd's definition of a transitive dependency is modified in the fashion just discussed.

Definition 2.1. Let R be a relation scheme and Σ a set of FDs which apply to it. R is in *third normal form (3NF)* if for every nontrivial FD $X \rightarrow A$ in Σ^+, either X is a superkey or A is a prime attribute.

We illustrate this definition by the following well known example [Beeri and Bernstein 1979; Date 1990].

Example 2.5. Let $R = \{STUDENT,\ COURSE,\ TEACHER\}$ and $\Sigma = \{STUDENT\ COURSE \rightarrow TEACHER, TEACHER \rightarrow COURSE\}$. An example of a relation defined over R is illustrated in Figure 2.3. The candidate keys are $STUDENT\ COURSE$ and $STUDENT\ TEACHER$. R is in 3NF because $STUDENT\ COURSE$ in the FD $STUDENT\ COURSE \rightarrow TEACHER$ is a superkey, and $COURSE$ in the FD $TEACHER \rightarrow COURSE$ is a prime attribute. □

STUDENT	COURSE	TEACHER
Jones	Physics	Newton
Walker	Physics	Maxwell
Jones	Maths	Hilbert
Smith	Physics	Newton

Figure 2.3. A relation defined on a 3NF scheme

From a computational perspective, testing a relation scheme for 3NF is computationally difficult since it has been shown to be NP-complete [Jou and Fischer 1983; Lucchesi and Osborn 1978].

2.6.2. Boyce-Codd Normal Form (BCNF)

Boyce-Codd normal form (*BCNF*) was introduced by Codd to overcome deficiencies in 3NF which may arise in the case where there are candidate keys which overlap [Codd 1974]. We now present a formal definition.

Definition 2.2. Let R be a relational scheme and let Σ be a set of FDs and MVDs which apply to it. R is in *Boyce-Codd normal form (BCNF)* if for every nontrivial FD $X \rightarrow A \in \Sigma^+$, X is a superkey.

Again, we illustrate this definition by an example taken from Date's book [Date 1990].

Example 2.6. Let $R = \{STUDENT, COURSE, POSITION\}$ with the meaning that a tuple $<s, c, p>$ defined over this scheme represents the information that in an examination for a course c, student s is ranked at position p. If one also imposes the constraints that each student in a course receives one position, and that no two students have the same position in a course, then the following FDs apply to the relation scheme - {*STUDENT COURSE* $\rightarrow$ *POSITION*, *COURSE POSITION* $\rightarrow$ *STUDENT*}. The candidate keys are *STUDENT COURSE* and *COURSE POSITION*. *R* is in BCNF since every nontrivial FD in Σ^+ must contain either *STUDENT COURSE* or *COURSE POSITION* in the left-hand side of the dependency (see Lemma 3.1 in the next chapter). An example of a relation defined over *R* is illustrated in Figure 2.4. □

STUDENT	COURSE	POSITION
Jones	Maths	1
Smith	Maths	2
Jones	Physics	4
Allan	Physics	1
Smith	Physics	2

Figure 2.4. A relation defined on a scheme that is in BCNF

We note that the relation scheme in Example 2.5 is not in BCNF because *TEACHER* in the dependency *TEACHER* $\rightarrow$ *COURSE* is not a superkey. It has been shown [Vincent and Srinivasan 1994b] that the situation in which a relation scheme can be in 3NF but not in BCNF can only occur when there is a pair of overlapping candidate keys. We note that the converse of this is not valid since the relation scheme in Example 2.6 has overlapping candidate keys yet is in BCNF.

2.6.3. Fourth Normal Form (4NF)

Fourth normal form (*4NF*) was defined in a paper by Fagin [Fagin 1977c] in which MVDs were also presented for the first time. 4NF was proposed as a generalisation of BCNF in order to cater for the case where the constraints are generalised to include both FDs and MVDs [Fagin 1977c]. The following definition is taken from Fagin's paper.

Definition 2.3. Let R be a relation scheme and Σ a set of MVDs and FDs that apply to R. R is in *fourth normal form* (*4NF*) if for every nontrivial MVD $X \rightarrow\rightarrow Y \in \Sigma^+$, X is a superkey.

Fagin also showed that if a relation scheme is in 4NF with respect to a set Σ of FDs and MVDs, then it is also in BCNF with respect to the FDs implied by Σ and so 4NF is a stronger condition than BCNF. In the following example taken from a paper by Zaniolo and Melankoff [Zaniolo and Melankoff 1982], a relation scheme is presented which is BCNF with respect to the FDs implied by Σ but is not in 4NF.

Example 2.7. Let $R = \{DAY, TIME, GROUP\}$ where a tuple $<d, t, g>$ defined over the scheme represents the information that *GROUP* g meets on *DAY* d at *TIME* t in an organisation's conference room. Only one group is allowed to meet in the room at any one time and so this constraint is represented by the FD $DAY\ TIME \rightarrow GROUP$. If one also imposes the additional constraint that a group may meet several times in a day, but these times are the same for every day that the group meets, then this constraint is represented by the MVD $GROUP \rightarrow\rightarrow TIME$. An example of a relation satisfying the constraints is shown in Figure 2.5. It can easily be verified that no other nontrivial FDs are implied by the set of constraints and so R is in BCNF since the only candidate key is *DAY TIME*. However, R is not in 4NF because *GROUP* is not a superkey in the MVD $GROUP \rightarrow\rightarrow TIME$. □

DAY	TIME	GROUP
Mon	9.00	G1
Wed	12.00	G1
Wed	9.00	G1
Mon	12.00	G1
Mon	2.00	G2
Tue	12.00	G3

Figure 2.5. A relation defined on a scheme that is in BCNF but not in 4NF

There is a divergence of opinion as to whether a set of constraints such as the one presented in the previous example represents a situation which can exist in the 'real-world'. For example, Nakamura and Chen [Nakamura and Chen 1981] argued that the previous example is essentially pathological and, apart from examples with a similar dependency structure, they proved that if a relation is in 4NF but not in BCNF then the only candidate key is the whole relation scheme. This also raises the more general issue of whether some sets of dependencies involving complex interactions only exist in the abstract sense, and do not occur in real-world applications. Once again there is a divergence of opinion on this question. Sciore, for example, argued that only a certain class of MVDs, called *conflict-free MVDs*, occur in practical examples [Lien 1982; Sciore 1981]. This was later questioned by Beeri and Vardi who proposed a real-world example where the set of MVDs is not conflict-free [Beeri and Vardi 1981a]. Later still, Ling [Ling 1985] examined the example proposed by Beeri and Vardi in an *entity-relationship* framework [Batini et al. 1991; Chen 1976] and argued that their example was incorrectly specified.

Our perspective is that while some complex dependency interactions seem to occur rarely in practice, it is an unprovable claim to state that they can never occur. What is probably less in dispute is that a set of dependencies can be incompletely specified and certain database design difficulties can be overcome if additional dependencies, which don't change the original semantics, are added to the set of dependencies. The problem of how to add dependencies to obtain a better database design without altering the semantics of the original set has been investigated by several researchers [Beeri and Kifer 1986a; Beeri and Kifer 1986b; Beeri and Kifer 1987; Kandzia and Manglemann 1980; Sciore 1983a]. We also note that several investigations have been conducted into the issue of understanding precisely the real-world interpretation of MVDs [Beeri and Vardi 1981a; Kambayashi et al. 1979; Katsuno 1981; Kent 1981; Ling 1985; Sciore 1981].

2.6.4. (3,3)NF

Historically, the next normal form to be defined was a normal form proposed by Smith called *(3,3)NF*. It was introduced as an improvement to 3NF for the case where an FD may hold in a subset of an attribute domain, but not in the full domain [Smith 1978]. To illustrate this concept, we reproduce the example from Smith's paper.

Example 2.8. Let $R = \{E\#, TYPE, PERCENTAGE\_TIME, PAY\}$. *E#* represents the identifier of an employee, *TYPE* represents the classification of an employee which may be "hourly employee" or "salaried employee", *PERCENTAGE_TIME* represents the fraction of time that an employee works and *PAY* represents the pay that an employee receives. An example of a relation defined over *R* is illustrated in Figure 2.6. The only FD is $E\# \rightarrow TYPE\ \ PERCENTAGE\_TIME\ \ PAY$. In particular it can be seen from Figure 2.6 that there is no FD from *PERCENTAGE_TIME* to *PAY*. However, there is also a constraint which states that for hourly employees only, the *PERCENTAGE_TIME* determines the *PAY*. In other words, while there is no FD from *PERCENTAGE_TIME* to *PAY* in the relation as a whole, there is an FD from *PERCENTAGE_TIME* to *PAY* in a subset of the rows (those who are hourly employees). It is easily seen that this FD causes difficulties in the relation, such as redundancy, although the relation scheme is in BCNF. The relation scheme is not in what Smith refers to as (3,3)NF. □

E#	TYPE	PERCENTAGE _TIME	PAY
E1	hourly	50	200
E2	hourly	75	400
E3	hourly	75	400
E4	hourly	100	500
E5	hourly	100	500
E6	salaried	100	650
E7	salaried	100	550
E8	salaried	50	400
E9	salaried	50	550

Figure 2.6. A relation defined on a scheme that is not in (3,3)NF

While Smith originally considered (3,3)NF as an extension of 3NF, the same arguments he used also apply to the case of MVDs and so one can extend the (3,3)NF definition to MVDs as follows.

Definition 2.4. A relation scheme R is in *(3,3)NF* if for every selection condition defined on a relation $r(R) \in \mathrm{SAT}(\Sigma)$, the only nontrivial MVDs which hold in the rows selected must be of the form $X \rightarrow\rightarrow Y$ where X is a superkey.

An interesting aspect of (3,3)NF is that in contrast to all the other normal forms discussed in this chapter with the exception of DK/NF (see later), the conversion of a relation scheme which is not in (3,3)NF to a set of schemes in (3,3)NF requires the application of horizontal decompositions. This is in contrast to what occurs for the other normal forms where only vertical decompositions are employed. For instance, to convert the relation scheme in Example 2.8 to (3,3)NF one first splits the scheme horizontally into an hourly employee relation with attributes *E#*, *PERCENTAGE_TIME* and *PAY* and another relation, salaried employee, with attributes *E#*, *PERCENTAGE_TIME* and *PAY*. The hourly employee scheme is then split vertically into two schemes, one containing the attributes *E#* and *PERCENTAGE_TIME* and the other the attributes *PERCENTAGE_TIME* and *PAY*. The possibility of improving a database design by horizontal decompositions has also been considered by others [DeBra and Paradaens 1982; DeBra and Paredaens 1983; DeBra and Paredaens 1990; Delobel 1978; Fagin 1979; Furtado 1981; Paredaens et al. 1989].

2.6.5. Projection-Join Normal Form (PJNF)

Projection-join normal form (*PJNF*) was introduced by Fagin [Fagin 1979] for the case where the constraints contain JDs.

Definition 2.5. Let R be a relation scheme and let Σ be a set of FDs and JDs which apply to R (any MVD is treated as a JD). R is in *projection-join normal form* (*PJNF*) if every relation $r(R)$ which satisfies Σ_K also satisfies Σ.

The following example illustrates this definition.

Example 2.9. Let R = {*SUPPLIER, PART, PROJECT*}. A tuple <*s, p, j*> in a relation defined over R represents the information that *SUPPLIER s* supplies *PART p* to *PROJECT j*. Suppose also that the JD constraint *[*SUPPLIER PART, SUPPLIER PROJECT, PART PROJECT*] applies to R. Then R is not in PJNF since the relation shown in Figure 2.7 satisfies Σ_K but not the JD because the relation doesn't contain the tuple <*s1*, *p1*, *j1*>. □

SUPPLIER	PART	PROJECT
s1	p1	j2
s2	p1	j1
s1	p2	j1

Figure 2.7. A relation defined a scheme that is not in PJNF

A recent paper by Date and Fagin [Date and Fagin 1992] has helped to clarify the relationship between 3NF, BCNF, 4NF and PJNF. They proved that if every candidate key is simple (contains only a single attribute), then every relation scheme that is in 3NF is also in PJNF (and hence in BCNF and 4NF as well). In other words, the only situation where the normal forms do not coincide is when a relation scheme has a complex key structure.

2.6.6. Elementary Key Normal Form (EKNF)

Another normal form that is stronger than 3NF yet weaker than BCNF, called *elementary key normal form* (*EKNF*), was defined by Zaniolo [Zaniolo 1982]. The following definitions are taken from this work.

Definition 2.6. Let Σ be a set of FDs and $X \to A$ an FD in Σ. An FD $X \to A$ is *elementary* with respect to Σ if there doesn't exist a nontrivial FD $X' \to A$ in Σ^+ such that $X' \subset X$.

A set of attributes K is an *elementary key* if for some attribute A, $K \to A$ is an elementary FD. An attribute which belongs to some elementary key is called an *elementary key attribute*.

A relation scheme R is in *elementary key normal form* (*EKNF*) if for every nontrivial FD $X \to A$ in Σ, either X is a superkey or A is an elementary key attribute.

The following example demonstrates a relation scheme which is in 3NF but not in EKNF [Zaniolo 1982].

Example 2.10. Let $R = \{DEPT, MNGR, ACC\#\}$ where $DEPT$ represents the name of a department, $MNGR$ represents the name of the manager of the department and $ACC\#$ represents an account used by the department. Suppose also that the following set of constraints apply - $\{DEPT \rightarrow MNGR, MNGR \rightarrow DEPT\}$. An example of a relation defined over this scheme is shown in Figure 2.8. Then the candidate keys are $DEPT\ ACC\#$ and $MNGR\ ACC\#$. Neither of these candidate keys is elementary because neither of the FDs $DEPT\ ACC\# \rightarrow MNGR$ nor $MNGR\ ACC\# \rightarrow DEPT$ is an elementary FD and so there are no elementary key attributes. Hence R is not in EKNF since if one considers the FD $DEPT \rightarrow MNGR$, then $DEPT$ is not a candidate key nor is $MNGR$ an elementary key attribute. However, R is in 3NF since the right-hand sides of both FDs are prime attributes. □

DEPT	MNGR	ACC#
Accounts	Allan	1
Accounts	Allan	2
Sales	Bloggs	1
Sales	Bloggs	3

Figure 2.8. A relation defined on a scheme that is in EKNF but not 3NF

The next normal form to be defined was an improvement of 3NF, appropriately called *improved 3NF* [Ling et al. 1981]. Unlike all the other normal forms discussed in this chapter which consider only single relation scheme, improved 3NF attempts to remove difficulties which can occur across multiple relation schemes. Since only single relation schemes are being considered in this work, improved 3NF won't be discussed further.

2.6.7. Domain-Key Normal Form (DK/NF)

The last normal form that we discuss is *domain-key normal form* (*DK/NF*) due to Fagin [Fagin 1981]. Although historically defined before EKNF, DK/NF is essentially the ultimate normal form and thus it has been left until the end. We will give only a brief overview of DK/NF here; a more complete discussion, especially that of the enforcement of key uniqueness, is contained later in Chapter 4.

The motivation for DK/NF is based on the desirability of guaranteeing, after an update to a relation, the satisfaction of all general constraints on a relation by enforcing only the satisfaction of the primitive constraints, where a primitive constraint can be either a *key dependency* (the restriction that there be no duplicates for certain sets of attributes) or a *domain dependency* (the restriction that an attribute value lies in a specific set). Since primitive constraints can be, and are [Date 1990], easily enforced in relational database software, Fagin thus considered that a desirable property of a relation scheme is that the satisfaction of every constraint on a relation defined over the scheme be guaranteed if the relation satisfies the primitive constraints. DK/NF is then defined as follows [Fagin 1981].

Definition 2.7. Let R be a 1NF relation scheme and let Γ be the set of domain dependencies and key dependencies of the schema. R is in *domain-key normal form* (*DK/NF*) if $\Gamma \Rightarrow \sigma$ for every constraint σ.

We illustrate the definition with an example [Fagin 1981].

Example 2.11. Let $R = \{EMP\#, STATUS, SALARY\}$ and let the constraints be:

DOM($EMP\#$) = {n: n is a six-digit integer}

DOM($STATUS$) = {$0, 1$}

DOM($SALARY$) = {n: $10{,}000 \leq n \leq 100{,}000$}

KEY($EMP\#$)

$\forall t((t[STATUS] = 0) \Rightarrow (t[SALARY] \leq 50{,}000)$

R is not in DK/NF since the relation shown in Figure 2.9 satisfies the domain and key constraints but violates the other constraint that an employee with a status of *0* cannot have a salary greater than *$50,000*. □

EMP#	STATUS	SALARY
329461	1	73000
141592	0	37000
272828	1	46000
141421	0	57000

Figure 2.9. A relation not in DK/NF

Fagin also proved that provided that the domains of attributes are sufficiently large, then DK/NF implies all the other normal forms so far defined. The relationship between the different normal forms is illustrated diagrammatically in Figure 2.10.

$$\begin{array}{c} \text{DK/NF} \\ \Downarrow \\ \text{PJNF} \\ \Downarrow \\ \text{(3,3)NF} \\ \Downarrow \\ \text{4NF} \\ \Downarrow \\ \text{BCNF} \\ \Downarrow \\ \text{EKNF} \\ \Downarrow \\ \text{3NF} \\ \Downarrow \\ \text{2NF} \\ \Downarrow \\ \text{1NF} \end{array}$$

Figure 2.10. The relationship between the normal forms

2.7. DESIRABLE DATABASE DESIGNS

We now briefly mention some other related issues which, although outside the scope of this thesis, are related to normalisation and important in relational database design. Essentially these issues are derived from the following central problem: starting with a relational scheme that's not normalised, how does one decompose it into a set of relational schemes that not only have the desirable property that the individual schemes are normalised, but also have the property that the semantics of the original scheme are in some sense preserved by these new schemes? While there have been many subtly different proposals for defining precisely what is meant by the semantics being preserved in a decomposition [Arora and Carlson 1978; Beeri et al. 1978; Beeri et al. 1981; Maier et al. 1980; Rissanen 1977; Rissanen 1982], most are based on the notions of *information preservation* and *dependency preservation*.

The *information preservation* condition requires that for any relation defined over the original scheme, the relation is equal to the join of its projections onto the decomposed schemes. In other words, the decomposition must be a JD. The *dependency* *preservation* condition essentially requires that any relation satisfying the constraints in the original relation implies that the decomposed relations also satisfy their individual constraints (which are the projections of the initial dependencies on the individual schemes), and conversely if the projected relations satisfy their individual constraints then their join satisfies the original constraints. Algorithms have been developed for testing whether a set of relation schemes satisfies either the information preservation or dependency preservation conditions [Aho et al. 1979a; Beeri and Honeyman 1981]. We now review some of the methods and results pertaining to the design of normalised schemes with the desirable properties just discussed.

In the case of FD constraints, the two most widely used techniques for achieving desirable database designs are the *synthesis method* and the *analysis method*[3] (also referred to as the decomposition method). The synthesis method, originally developed by Bernstein [Bernstein 1976], starts with a reduced set of FDs (also called a minimal cover) and generates relation schemes directly from the FDs in the reduced set. This method has the desirable property that it has been shown to generate schemes which are in 3NF as well as being information preserving and constraint preserving [Bernstein 1976; Biskup et al. 1979]. It was also later established that the schemes generated from the synthesis method in fact satisfy the stronger EKNF condition [Zaniolo 1982]. However, the synthesis method does not always generate BCNF relation schemes and for this case the analysis method is used. The technique used in the analysis method is to successively split a relation scheme whenever a BCNF violation occurs until a set of BCNF schemes is produced. It can be shown that the analysis method is information preserving but is not in general constraint preserving since, for some sets of FDs, no decomposition exists which is BCNF and both information and dependency preserving [Beeri and Bernstein 1979].

---

[3]This terminology has been adopted from Atzeni and De Antonellis [Atzeni and DeAntonellis 1993].

In the case of the constraints containing only MVDs, all the published algorithms for generating 4NF schemes are based on the analysis method [Beeri and Kifer 1986b; Fagin 1977a; Fagin 1977c; Grahne and Raiha 1983; Lien 1981; Sciore 1981; Zaniolo and Melankoff 1981; Zaniolo and Melankoff 1982]. However, as for BCNF, a 4NF decomposition which is both information preserving and dependency preserving doesn't exist unless the MVDs are of a restricted type, referred to as *conflict-free MVDs* [Lien 1982; Sciore 1981]. Under this condition, Lien proved that a necessary and sufficient condition for the existence of an information preserving, dependency preserving 4NF decomposition is that the set of MVDs be conflict-free [Lien 1982]. He also showed that for this case, the decomposition is unique. Somewhat surprisingly, the derivation of necessary and sufficient conditions for the existence of a desirable 4NF decompositions in the more general case where the constraints contain both MVDs and FDs was not solved until very recently by Yuan and Ozsoyoglu [Yuan and Ozsoyoglu 1986; Yuan and Ozsoyoglu 1987; Yuan and Ozsoyoglu 1992a; Yuan and Ozsoyoglu 1992b]. The difficulty arises in this case because MVDs and FDs have different semantics, and so simply treating an FD as an MVD and requiring the resulting set of MVDs to be conflict-free does not result in a necessary and sufficient condition. The required condition involves what is called an *extended conflict-free* set of dependencies, which is essentially the conflict-free property applied to a special set of MVDs implied by the original set of FDs and MVDs, called the *envelope set*. Roughly speaking, the envelope set is the set of MVDs which do not have FD counterparts and is based on similar ideas to the notion of a pure MVD defined in Chapter 2. For an extend conflict-free set of dependencies, Yuan and Ozsoyoglu also showed that the relation schemes resulting from the decomposition satisfied another condition, namely that the set of schemes is *acyclic*. Acyclicity is a property that results from viewing a set of relation schemes as a hypergraph and it is known that acyclic schemes have several desirable processing properties [Beeri et al. 1983; Fagin 1983].

In concluding this section on desirable database designs, we note that there exist other frameworks for viewing the properties of desirable designs than the one introduced at the start of this section - namely that the database design process is viewed as a decomposition of an initial single relational scheme into a set of more desirable relation schemes. This framework, generally called the *pure universal relation assumption*, has generated considerable controversy in the research literature [Atzeni and Parker 1982; Fagin et al. 1982; Kent 1981; Maier et al. 1986; Maier et al. 1984; Sciore 1980; Ullman 1982; Ullman 1983a; Ullman 1987]. Aside from some of the philosophical arguments over the pure universal relation assumption, there are also practical difficulties in the use of the assumption since not all sets of relations are the projections of a single relation and testing this property has been shown to be NP-complete [Honeyman et al. 1980; Maier et al. 1981]. A popular alternative approach is based on a variant of the universal relation assumption, called the weak instance approach [Honeyman 1982; Sagiv 1981], which is less restrictive than the pure instance assumption. Mendelzon derived several results concerning desirable decompositions under this approach [Mendelzon 1984]. More generally, Hull looked at equivalence between two sets of relation schemes, rather than between a single scheme and a set of schemes, without using any form of the universal relation assumption [Hull 1986].

CHAPTER 3

REDUNDANCY AND NORMAL FORMS

3.1. INTRODUCTION

In most database texts [Date 1990; Ullman 1988a; Vossen 1990], one of the main intuitive justifications proposed for normalisation, apart from the avoidance of update anomalies which we will investigate in later chapters, is the elimination of redundancy. In this chapter we address the issue of formally defining the intuitive notion of redundancy and then derive results concerning necessary and sufficient conditions for the absence of redundancy in relations and relation schemes. We also note that the results of this chapter are reported in the literature [Vincent 1991; Vincent and Srinivasan 1992a; Vincent and Srinivasan 1992b; Vincent and Srinivasan 1994c]. The content and structure of this chapter will now be outlined.

In Section 3.2, the property of redundancy is analysed based on the idea of viewing the set of attributes in an FD or MVD constraint as being a *fact* or atomic unit of information. A relation is said to contain redundancy if it has two or more tuples which are identical on a fact and a relation scheme is defined to be redundant if there exists a *legal relation* (satisfies the set of constraints) defined over the scheme which contains redundancy. To be more precise, we define three types of redundancy in a relation scheme since there are three possible choices for the set of facts. For the first type of redundancy (called RED_1), the set of facts is chosen to be the sets of attributes of the FDs and MVDs in the set of dependencies, Σ, supplied by the database designer; for the second (called RED_2), the set of attributes in the MVD $X \rightarrow\rightarrow R - XY$ corresponding to an MVD $X \rightarrow\rightarrow Y$ in Σ is also defined to be a fact; and for the last one (called RED_3), the set of facts is defined to contain the sets of attributes of all the nontrivial FDs and MVDs which are logically implied by Σ. The motivation for defining RED_2 is that the complementary rule for MVDs (rule A4 in Chapter 2), which has no counterpart in the inference rules for FDs, implies that the MVD $X \rightarrow\rightarrow Y$ is satisfied in a relation if and only if the MVD $X \rightarrow\rightarrow R - XY$ is also satisfied, so there is no real basis for considering either one of these MVDs as being more important than the other and thus we consider both as facts.

Preliminary lemmas needed for proving the main results of this chapter are derived in Section 3.3. The main results obtained in this section are that in the definitions of 3NF, BCNF and 4NF, equivalent definitions are obtained if the requirement that all the dependencies in Σ^+ have the desired property is replaced by the requirement that all the dependencies in Σ have the desired property. While several proofs of this result have already been given for the 3NF and BCNF cases [Atzeni and DeAntonellis 1993; Vossen 1990], we present, to our knowledge for the first time, a proof for the 4NF case. In fact, we give two proofs of this result which provide different insights into the nature of normal forms.

In Sections 3.4, 3.5 and 3.6, necessary and sufficient conditions are derived for the absence of the three types of redundancy in a relation scheme for three separate cases - the constraints contain only FDs, the constraints contain only MVDs, and the constraints contain both FDs and MVDs. In Section 3.4 we show that, for the FD case, RED_1, RED_2 and RED_3 are equivalent conditions on a relation scheme and BCNF is equivalent to an absence of either type of redundancy. RED_1, RED_2 and RED_3 are also shown in Section 3.5 to be equivalent conditions on a relation scheme when the only constraints are MVDs. It is also established in the same section that 4NF is equivalent to an absence of any of the redundancy types. Section 3.6 contains perhaps the most surprising result of this chapter. We show that for the case where the constraints includes both FDs and MVDs, RED_2 and RED_3 are equivalent and 4NF is equivalent to an absence of either. However, we also prove that an absence of RED_1 in a relation scheme is a weaker condition than 4NF and derive an equivalent syntactic characterisation of those schemes which are not RED_1. We then prove that if the MVDs in the set of constraints are restricted to pure MVDs (see Chapter 2), then RED_1 is equivalent to RED_2 and RED_3 (and hence its absence is also equivalent to 4NF).

3.2. THE DEFINITION OF REDUNDANCY

The approach taken in this chapter to defining redundancy is based on viewing FDs and MVDs as not only integrity constraints on a relation, but also as representing the fundamental units of information for retrieving and updating the data in a relation. For example, suppose one is given the relation scheme $\{SCODE, SNAME, TEACHER, TEXT\}$ with the meaning that a tuple $<a, b, c, d>$ over the scheme represents the information that a subject with code a and name b is taught by a teacher c and uses a text book d. If the set of dependencies which apply to the scheme is $\{SCODE \rightarrow SNAME, SCODE \rightarrow\rightarrow TEXT, SCODE \rightarrow\rightarrow TEACHER\}$, then we consider the facts to be the sets $\{SCODE, SNAME\}$, $\{SCODE, TEXT\}$, $\{SCODE, TEACHER\}$.

This interpretation of the semantics of the information stored in a relation was implicit in the original study of normalisation by Codd [Codd 1972], and has since been used in many aspects of database theory including database design [Bernstein 1976; Biller 1979; Biskup et al. 1979; Hall et al. 1976; LeDoux and Parker 1982; Nijssen 1977; Nijssen 1979; Nijssen and Halpin 1989], the justification for normalisation [Bernstein and Goodman 1980; Biskup 1989; Chan 1989; LeDoux and Parker 1982; Vossen 1988], semantics of database updates [Desai et al. 1986; Desai et al. 1987; Fagin et al. 1986; Fagin et al. 1983], universal relation databases [Korth et al. 1984; Maier et al. 1986; Maier and Ullman 1983; Maier et al. 1984; Maier and Warren 1982; Sciore 1980] and the equivalence of database decompositions [Beeri et al. 1981]. We define a relation scheme to be redundant if there exists a legal relation defined over the scheme which has two or more tuples which are identical on a fact. For instance, the relation scheme just presented is redundant because the relation shown in Figure 3.1 satisfies the constraints and has two tuples which are identical on {*SCODE*, *TEACHER*} (and also two tuples which are identical on the other facts).

SCODE	SNAME	TEACHER	TEXT
s_1	n_1	p_1	t_1
s_1	n_1	p_2	t_2
s_1	n_1	p_1	t_2
s_1	n_1	p_2	t_1

Figure 3.1. A relation containing redundancy

A more formal rationale for this definition of redundancy [Beeri et al. 1978] is the result due to Fagin [Fagin 1977c] which states that an MVD $X \rightarrow\rightarrow Y$ holds in a relation r if and only if r is equal to the join of its projections onto XY and XZ. Thus if two tuples in r are identical on XY, then the same information is represented by a single tuple when projections are taken (and thus duplicates are removed). So the reason for redundancy being undesirable is that it results in the wastage of secondary storage space by duplicating the same unit of information and also, as will be seen later in Chapter 6, can lead to associated anomalies when a relation is updated.

The last issue that has to be addressed before formally defining redundancy is the subtle point of determining what to use as the set of facts. From an intuitive viewpoint, this is not as easy to decide as may first appear and we propose three possibilities and investigate the implications of each. The first is to simply allow the set of facts to be the sets of attributes in all the nontrivial FDs and MVDs in a user-supplied set of dependencies Σ. The second is to recognise the symmetrical nature of MVDs and so allow the set of attributes in any MVD that can be derived from any MVD in Σ and inference rule A4 (see Chapter 2) to also be a fact. The last possibility is to include derived dependencies and allow the set of attributes in any nontrivial FD or MVD that is implied by Σ to be a fact. Intuitively, one would expect the redundancy property to be independent of which of these sets of facts is chosen but, as will be seen later, in general this is not the case and, even when it is, the proof of this fact is by no means immediate. We now present formal definitions of the different types of redundancy.

Definition 3.1. Let R be a relation scheme and let Σ be a set of FDs and MVDs which apply to R. A relation scheme R is *redundant*$_1$ (abbreviated subsequently to RED_1) if there exists a relation $r(R)$ in SAT(Σ) and a nontrivial dependency $X \rightarrow\rightarrow Y$ or $X \rightarrow Y$ in Σ and at least two distinct tuples $t_1, t_2 \in r$ such that $t_1[XY] = t_2[XY]$.

Definition 3.2. Let R be a relation scheme and let Σ be a set of FDs and MVDs which apply to R. Define the set Σ' by $\Sigma' = \Sigma \cup \{X \rightarrow\rightarrow R - XY \mid X \rightarrow\rightarrow Y \in \Sigma\}$. A relation scheme R is *redundant*$_2$ (abbreviated subsequently to RED_2) if there exists a relation $r(R)$ in SAT(Σ) and a nontrivial dependency $X \rightarrow\rightarrow Y$ or $X \rightarrow Y$ in Σ' and at least two distinct tuples $t_1, t_2 \in r$ such that $t_1[XY] = t_2[XY]$.

Definition 3.3. Let R be a relation scheme and let Σ be a set of FDs and MVDs which apply to R. A relation scheme R is *redundant*$_3$ (abbreviated subsequently to RED_3) if there exists a relation $r(R)$ in SAT(Σ) and a nontrivial dependency $X \rightarrow\rightarrow Y$ or $X \rightarrow Y$ in Σ^+ and at least two distinct tuples $t_1, t_2 \in r$ such that $t_1[XY] = t_2[XY]$.

We now illustrate the previous definitions by an example.

Example 3.1. Let $R = \{A, B, C\}$ and $\Sigma = \{A \rightarrow\rightarrow B, B \rightarrow A, B \rightarrow C\}$. It can be easily verified that the only candidate key in R is B and thus R is not in 4NF because of the MVD $A \rightarrow\rightarrow B$. From rule A4 and $A \rightarrow\rightarrow B$, it follows that $A \rightarrow\rightarrow C$ is in Σ' and hence is also in Σ^+. Then R is both RED_2 and RED_3 since the relation r shown in Figure 3.2 is in SAT(Σ) and the two tuples are identical on AC. However, r is not redundant on Σ and, since every dependency in Σ contains the candidate key B, no relation defined over R can have duplicates on a dependency in Σ and so R is not RED_1.

□

r

A	B	C
a_1	b_1	c_1
a_1	b_2	c_1

Figure 3.2. A relation which is RED_2 and RED_3

It should be noted that since $\Sigma \subseteq \Sigma' \subseteq \Sigma^+$, an immediate consequence of the redundancy definitions are the following implications: a relation scheme R is $\text{RED}_1 \Rightarrow R$ is $\text{RED}_2 \Rightarrow R$ is RED_3. However, as demonstrated in Example 3.1, some of the converses do not hold and in particular a relation scheme can be RED_2 and RED_3 but not RED_1.

Another issue that arises from the definitions of redundancy is whether the property of a relation scheme being RED_1 or RED_2 is invariant under the replacement of the set of dependencies by an equivalent set. Intuitively, it is desirable that the properties do not change and we will investigate this issue in later sections of this chapter.

3.3. CERTAIN PROPERTIES OF NORMAL FORMS

We now derive several basic properties of normal forms which will be used later in the main results of this and later chapters. The following lemma relates the structure of the dependencies in Σ^+ to those in Σ.

Lemma 3.1. *If R is a relation scheme and Σ is a set of MVDs and FDs that apply to R, then for any nontrivial dependency $X \rightarrow\rightarrow W$ or $X \rightarrow W$ in Σ^+ there exists a nontrivial dependency $X' \rightarrow\rightarrow Y$ or $X' \rightarrow Y$ in Σ such that $X' \subseteq X$.*

Proof. In the case of the dependency in Σ^+ being an MVD $X \rightarrow\rightarrow W$, write the MVD as the JD $*[XW, XZ]$ where $Z = R - XW$ and form the tableau $T_{\mathbf{R}}$ as described in Chapter 2 and then let $T^* = chase_{\Sigma}(T_{\mathbf{R}})$. Using Lemma 2.3, since $X \rightarrow\rightarrow W \in \Sigma^+$ there has to be a row in T^* that contains only distinguished variables. Then since $X \rightarrow\rightarrow W$ is nontrivial, $R - XW \neq \emptyset$ and $R - XZ \neq \emptyset$ and so it follows from the construction of $T_{\mathbf{R}}$ that neither row in it contains only distinguished variables and thus there has to be an application of a J-rule or an F-rule to $T_{\mathbf{R}}$ during the chase. However, by definitions of the F-rule and the J-rule for an MVD, one can only apply them to two rows that are identical on the left-hand sides of the corresponding dependency. Then, since the rows in $T_{\mathbf{R}}$ are identical only on the attributes in X, it follows that the rule that is applied to $T_{\mathbf{R}}$ must correspond to a dependency in Σ of the form $X' \rightarrow\rightarrow Y$ or $X' \rightarrow Y$ where $X' \subseteq X$. The dependency must also be nontrivial since trivial dependencies do not change a tableau.

In the other case of the dependency in Σ^+ being an FD $X \rightarrow W$, form the tableau T_X as described in Chapter 2 and let $T^* = chase_{\Sigma}(T_X)$. Then by Lemma 2.2, since $X \rightarrow W \in \Sigma^+$, each of the columns of $T^*[W]$ contains a single distinguished variable. However, since $X \rightarrow W$ is nontrivial, Y contains at least one attribute A that is disjoint from X and so $T_X[A]$ contains a distinguished variable and a nondistinguished variable and so there has to be at least one application of an F-rule or a J-rule to T_X. Since the rows in T_X are equal only on X, the same argument as for the MVD case applies and the result is established. □

We note that in this lemma, the dependency in Σ corresponding to the dependency in Σ^+ may not be of the same type. For example, if $R = \{A, B, C\}$ and $\Sigma = \{A \rightarrow B\}$

then an application of inference rules A4 and A8 (see Chapter 2) shows that $A \rightarrow\rightarrow C \in \Sigma^+$ but there is no MVD in Σ of the form $A \rightarrow\rightarrow Y$. Similarly, if $\Sigma = \{A \rightarrow\rightarrow C, C \rightarrow B\}$ then an application of inference rules A8 and A9 shows that $A \rightarrow B \in \Sigma^+$ but there is no dependency of the form $A \rightarrow Y$ in Σ. A different proof of this lemma is also possible using the properties of the algorithm, due to Beeri, for generating the dependency basis of a set of attributes [Beeri 1980]. We now use this lemma to establish the following main theorem. It shows that in testing for 4NF, it suffices to test only the dependencies in Σ rather than the dependencies in Σ^+ as required by the definition of 4NF (refer to Section 2.6.3). A similar result has also been established using different methods for BCNF in the case where the set of constraints contains only FDs [Vossen 1988; Vossen 1990] .

Theorem 3.2. *Let R be a relation scheme and let Σ be a set of FDs and MVDs which apply to R. R is in 4NF iff the left-hand side of every nontrivial dependency in Σ is a superkey.*

Proof.

If

Suppose to the contrary that R is not in 4NF. Then there exists a nontrivial MVD $X \rightarrow\rightarrow Y \in \Sigma^+$ such that X is not a superkey. By Lemma 3.1, there exists either a nontrivial MVD $X' \rightarrow\rightarrow Y$ or a nontrivial FD $X' \rightarrow Y \in \Sigma$ such that $X' \subseteq X$. However, a simple application of the inference rules shows that if X is not a superkey then X' is not a superkey, which contradicts the assumption that the left-hand side of every dependency in Σ is a superkey.

Only If

Automatic since $\Sigma \subseteq \Sigma^+$. □

The following simple corollary of this theorem provides another proof of the result that the property in Theorem 3.2 also holds for BCNF and FDs.

Corollary 3.3. *Let R be a relation scheme and let Σ be a set of FDs which apply to R. R is in BCNF iff the left-hand side of every nontrivial FD in Σ is a superkey.*

Proof. Immediate from the theorem and the well known result that 4NF implies BCNF [Fagin 1977c]. □

A different proof of Theorem 3.2 can be provided by using the following result due to Fagin which provides an alternative characterisation of 4NF in terms of key satisfaction [Fagin 1979]. This interpretation of normal forms will be investigated in more detail in a Chapter 4. For the sake of completeness, we present in the following lemma a simplified proof of Fagin's result.

Lemma 3.4. *Let R be a relation scheme and let Σ be a set of FDs and MVDs which apply to R. R is in 4NF iff every relation r(R) which is in SAT(Σ_k) is also in SAT(Σ).*

Proof.

Only If

Suppose that there is a relation which satisfies Σ_k but not Σ. Then there must exist either $X \rightarrow Y$ or $X \rightarrow\rightarrow Y \in \Sigma$ such that X is not a superkey. In the case of the dependency Σ being the MVD $X \rightarrow\rightarrow Y$ it immediately contradicts the assumption that R is in 4NF. Alternatively, if the dependency is the FD $X \rightarrow Y$ then by inference rule A8 (see Chapter 2) $X \rightarrow\rightarrow Y$ is also in Σ^+. Because X is not a superkey, $R - XY \neq \emptyset$ and so $X \rightarrow\rightarrow Y$ is nontrivial which again contradicts the assumption that R is in 4NF.

<u>If</u>

We shall show the contrapositive that if R is not in 4NF then there exists a relation which satisfies Σ_K but not Σ. Since R is not in 4NF, let $X \rightarrow\rightarrow Y$ be a nontrivial MVD in Σ^+ such that X is not a superkey. Construct a two tuple relation r for which the two tuples are identical on X and different elsewhere. Firstly, we claim that $r \in \text{SAT}(\Sigma_K)$. This follows from the definition of r and the fact that for any candidate key K, $K - X \neq \emptyset$ or else a simple application of the inference would imply the contradiction that X is a superkey. Next, we claim that r violates $X \rightarrow\rightarrow Y$. This follows because by definition of r the two tuples agree on X and disagree on Y, and they also disagree on $R - XY$ since $X \rightarrow\rightarrow Y$ is nontrivial and from the definition of r. It then follows from the definition of an MVD that r violates $X \rightarrow\rightarrow Y$. □

We are now in a position to provide an alternative proof of Theorem 3.2.

Theorem 3.2. *Let R be a relation scheme and let Σ be a set of FDs and MVDs which apply to R. R is in 4NF iff the left-hand side of every nontrivial dependency in Σ is a superkey.*

<u>If</u>

Suppose to the contrary that R is not in 4NF. Then by Lemma 3.5, there exists a relation $r(R)$ which satisfies Σ_K but violates Σ. However, for a dependency $X \rightarrow Y$ or $X \rightarrow\rightarrow Y$ to be violated, it has to be nontrivial and there has to be at least two tuples in r which are identical on X. Then since r satisfies Σ_K and the properties of a superkey, this implies that X cannot be a superkey which contradicts the assumption that the left-hand side of every nontrivial dependency in Σ is a superkey.

<u>Only If</u>

As for previous proof. □

We now turn our attention to 3NF and show that a similar result to Theorem 3.2 holds for the case of 3NF. It is also possible to prove the lemma by using the properties of the algorithm developed by Beeri for calculating the closure of a set of attributes [Beeri and Bernstein 1979].

Lemma 3.5. *Let R be a relation scheme and let Σ be a set of FDs which apply to R. R is in 3NF iff for every nontrivial FD $Y \rightarrow A \in \Sigma$, either Y is a superkey or A is a prime attribute.*

Proof.

Only If

Immediate.

If

Without loss of generality, we assume that the right-hand side of each FD in Σ is a single attribute since any FD $Y \rightarrow A_1 \ldots A_n$ is equivalent to the set $\{Y \rightarrow A_1, \ldots, Y \rightarrow A_n\}$ [Maier 1983]. We shall assume that the result of the lemma is not true, in other words there is a nontrivial FD $X \rightarrow A$ in Σ^+ where X is not a superkey and A is not prime and then derive a contradiction. Form the tableau T_X as described in Chapter 2 and let $T^* = chase_\Sigma(T_X)$. Then by Lemma 2.2, $\omega^*_X[A]$ must contain a distinguished variable, but since $X \rightarrow A$ is nontrivial, A is not a subset of X and so by definition of T_X, $\omega_X[A]$ is a nondistinguished variable so there must have been at least one application of the F-rule during the chase which changed the nondistinguished variable in $\omega_X[A]$ to a distinguished variable in $\omega^*_X[A]$. Let $Y \rightarrow A$ be the FD in Σ used in the application of the F-rule. By definition of the F-rule, both rows of the tableau at that stage must have contained a single variable in each of the Y-columns in order for the F-rule to be applied. But since the F-rule never changes a distinguished variable to a nondistinguished variable and T_X contains a distinguished variable in every column, the variables in the Y-columns when the F-rule was applied must have been distinguished variables. Again, since the chase doesn't change distinguished variables, this implies that the distinguished variables in the Y-columns will not be altered by subsequent F-rule applications and so each Y-column in T^* must contain only a single distinguished variable. This implies, by Lemma 2.2, that $X \rightarrow Y \in \Sigma^+$ which implies, by inference rule A3, that X is a superkey since Y is a super by assumption, which is a contradiction. □

A few observations on the computational implications of the previous results are appropriate at this point. In testing for BCNF or 4NF, by Theorem 3.2 and Corollary 3.3 it is sufficient to check that the left-hand side of every dependency in Σ is a superkey. Testing whether a single set of attributes is a superkey can be done in polynomial time on the number of attributes and the number of dependencies [Beeri and Bernstein 1979] and so testing the whole set of dependencies for the BCNF or 4NF property similarly takes polynomial time. This is more efficient than testing every dependency in Σ^+ since the number of dependencies in Σ^+ can be exponentially proportional to the number of dependencies in Σ [Maier 1983]. It should be noted that this result only applies to testing a single relation scheme and its set of dependencies. Instead, if a single relation scheme is decomposed into several relation schemes and one wishes to test if any of these resulting schemes are individually in BCNF or 4NF, then such a test is computationally expensive since projected dependencies can only be determined by generating Σ^+ [Beeri and Bernstein 1979]. We also note that while Lemma 3.5 leads to a more efficient method for determining if a single relation scheme is in 3NF than testing every FD in Σ^+, the method is still NP-complete because of the need to check if an attribute is prime [Beeri and Bernstein 1979; Lucchesi and Osborn 1978].

3.4. THE CASE OF FD CONSTRAINTS

Based on the definitions given in Section 3.2, in this section we derive the necessary and sufficient conditions for a relation scheme not to be RED_1 or RED_3 (since the constraints contain only FDs, RED_2 is equivalent to RED_1 and so is not considered separately). The main results are that RED_1 and RED_3 are equivalent and their absence in a relation scheme is equivalent to BCNF.

Before presenting the main theorem, we derive an important preliminary lemma which guarantees the existence of redundant relations under certain conditions.

Lemma 3.6. *Let R be a relation scheme and let Σ be a set of FDs and MVDs which apply to R. If X is a set of attributes that is not a superkey, then there exists a relation containing at least two tuples that satisfies Σ and for which all tuples are identical on X.*

Proof. Let T_X, ω_d, ω_X, T^*, ω_d^* and ω_X^* be as defined in Chapter 2. Also, let ρ be any one-to-one valuation of T^*. Such a valuation always exists because of the assumption of infinite domains for attributes and the fact that T^* contains a finite number of rows since the chase always terminates. The claim is that $\rho(T^*)$ is the relation required.

Firstly, by Lemma 2.1, $\rho(T^*)$ satisfies Σ. Secondly, T^*, and hence $\rho(T^*)$, must consist of more than one row. This follows because from Lemma 2.2, if T^* consists of one row then it must be ω_d, but since ω_d contains only distinguished variables, this would imply by Lemma 2.2 that X is a superkey contradicting the assumptions of the lemma.

Finally, we claim that for each attribute $A \in X$, $T^*[A]$ consists of a single distinguished variable and so all rows in T^* and thus $\rho(T^*)$ are identical on X. This follows from an inductive argument. Let T' represent the tableau at some stage of the chase and assume inductively that for all $A \in X$, $T'[A]$ consists of a single distinguished variable. Then if a J-rule is applied to T' to produce a new row ω', then by definition of the J-rule, for each attribute $B \in R$, there is a row ω in T' such that $\omega'[B] = \omega[B]$. So, by the induction hypothesis, this implies that for every attribute $A \in X$, $\omega'[A]$ will contain the same distinguished variable as $T'[A]$ and the inductive hypothesis is again true. Alternatively, if an F-rule is applied then the distinguished variable in each of the columns in X will remain unchanged since the F-rule does not change distinguished variables. Initially, by definition of T_X, each column in X contains a single distinguished variable, so the inductive principle applies and each column $A \in X$, $T^*[A]$ consists of a single distinguished variable and the result is proven. □

We note that while this lemma is sufficient for the purposes of this chapter, in fact we shall show in Chapter 5, by a different technique, that it can be strengthened to show that there is a relation of two tuples with the stated property. It is interesting to note, however, that when the dependencies are extended to include JDs, then the proof just given is still valid and so the lemma holds in this more general case. However, to our knowledge, it is not known if the stronger form of the lemma - that there is a relation of exactly two tuples - is valid in the more general case when JDs are included. We now use this lemma to establish the main theorem of this section.

Theorem 3.7. *Let R be a relation scheme and let Σ be a set of FDs which apply to R. The following are equivalent:*

(i) R is in BCNF;

(ii) R is not RED_1;

(iii) R is not RED_3.

Proof.

(i) ⇒ (ii)

Assume to the contrary that R is in BCNF but is not RED_1. Then by definition of RED_1 there exists a nontrivial FD $X \to Y \in \Sigma$ and a relation $r \in$ SAT(Σ) which has two or more tuples that are identical on XY. But since $r \in$ SAT(Σ), $r \in$ SAT(Σ_k) which implies that X cannot be a superkey and thus contradicts the assumption that R is in BCNF.

(ii) $\Rightarrow$ (iii)

The contrapositive, that RED_1 implies RED_3, follows automatically from the redundancy definitions and the fact that $\Sigma \subseteq \Sigma^+$.

(iii) $\Rightarrow$ (i)

We shall show the contrapositive that if R is not in BCNF then it is RED_1. Since R is not in BCNF, there exists a nontrivial FD $X \rightarrow Y$ in Σ^+ with X not a superkey. However, using Lemma 3.1, there must exist an FD $X' \rightarrow Y \in \Sigma$ with $X' \subseteq X$. Since X is not a superkey, X' is not a superkey and simple application of the inference rules shows that $X'Y$ is not a superkey. The conditions of Lemma 3.6 are satisfied and thus R is RED_1. □

We note that a simple corollary to this theorem is the result that a relation scheme is RED_1 with respect to one set of FDs if and only if it is RED_1 with respect to any equivalent set of FDs. This answers the question, for the case of FDs, of whether the RED_1 property is invariant under replacement of the set of dependencies by an equivalent set.

3.5. THE CASE OF MVD CONSTRAINTS

In this section the relationship between the three types of redundancy and 4NF is investigated for the case where the only constraints are MVDs. The main result derived is that the definitions of the three types of redundancy are equivalent and a scheme is in 4NF if and only if it is free of all three types of redundancy. Before proving these theorems, some preliminary results are first established.

Lemma 3.8. *If R is a relation scheme and Σ is a set of MVDs which apply to R, then the only candidate key in R is R itself.*

Proof. From a result due to Maier[4] [Maier 1983], the only FDs implied by a set of MVDs are trivial FDs. Suppose then there is a candidate key K such that $K \subset R$. This implies the contradiction that $K \rightarrow R - K$ is a nontrivial FD implied by Σ. □

Lemma 3.9. *Let Σ be a set of MVDs. If $X \rightarrow\rightarrow Y$ is a nontrivial MVD in Σ^+ and X is not a superkey, then XY is not a superkey.*

Proof. If XY is a superkey then it must contain a candidate key and so by Lemma 3.8, $XY = R$ contradicting the assumption that $X \rightarrow\rightarrow Y$ is nontrivial. □

Theorem 3.10. *R is in 4NF iff it is not RED_3.*

Proof.

Only If

As for Theorem 3.7.

If

Assume to the contrary that R is not RED_3 and not in 4NF. Since R is not in 4NF, there exists a nontrivial MVD $X \rightarrow\rightarrow Y$ in Σ^+ where X is not a superkey. Then by Lemma 3.9, XY is not a superkey and so by Lemma 3.6 R is RED_3. This is a contradiction and so R is in 4NF. □

Next, it will be shown that RED_1, RED_2 and RED_3 are equivalent for the case where the only dependencies are MVDs.

---

[4] Corollary of Theorem 8.11

Theorem 3.11. *If R is a relation scheme and Σ is a set of MVDs which apply to R, then the following are equivalent:*

(i) R is RED_1;

(ii) R is RED_2;

(iii) R is RED_3.

Proof. We shall show that (iii) ⇒ (ii) ⇒ (i). (i) ⇒ (ii) ⇒ (iii) follow directly from the definitions of redundancy.

(iii) ⇒ (ii)

If R is RED_3, then by Theorem 3.10 it is not in 4NF. So by Theorem 3.2 there exists a nontrivial MVD $X \rightarrow\rightarrow Y$ in Σ such that X is not a superkey. Then using Lemma 3.9, XY is not a superkey and so Lemma 3.6 implies that R is RED_2.

(ii) ⇒ (i)

Suppose R is RED_2. Then there is a nontrivial MVD $X \rightarrow\rightarrow Y \in \Sigma'$ and a relation of at least two tuples which is identical on XY. This implies that X cannot be a superkey since r satisfies Σ and hence Σ_k. If $X \rightarrow\rightarrow Y \in \Sigma$ then, by Lemma 3.9, XY cannot be a superkey and so by Lemma 3.6 R is RED_1. Alternatively, if $X \rightarrow\rightarrow Y \notin \Sigma$, then $X \rightarrow\rightarrow Z \in \Sigma$ where $Z = R - XY$. This is a nontrivial MVD since by assumption $X \rightarrow\rightarrow Y$ is nontrivial. So, by Lemma 3.9, XZ is not a superkey and then Lemma 3.6 implies that R is RED_1. □

3.6. THE CASE OF FD AND MVD CONSTRAINTS

In this section, we investigate the relationship between 4NF and the three types of redundancy in a relation scheme for the most general case where the set of constraints contains both FDs and MVDs. Firstly, two preliminary lemmas are presented before the main results of this section are derived.

Lemma 3.12. *If $X \rightarrow\rightarrow Y$ is a nontrivial MVD in Σ^+ such that XY is a superkey then $X \rightarrow R - XY$ is a nontrivial FD in Σ^+.*

Proof. If X is a superkey then the result is immediate. Assume then that X is not a superkey and let $Z = R - XY$. Since $X \rightarrow\rightarrow Z$ by rule A4 (refer to Chapter 2) and $XY \rightarrow Z$ because XY is a superkey, it follows that $X \rightarrow Z$ by rule A9 and because XY and Z are disjoint. The FD is also nontrivial by the definition of Z and because $R - XY$ is nonempty since $X \rightarrow\rightarrow Y$ is nontrivial. □

Lemma 3.13. *If $X \rightarrow Y$ is an FD such that XY is a superkey then X is a superkey.*

Proof. $X \rightarrow XY$ by rule A2 and $XY \rightarrow R$ since XY is a superkey, so using rule A3 this implies that $X \rightarrow R$. □

We now present one of the main results of the chapter which shows that the condition that a relation scheme is not RED_3 is equivalent to 4NF.

Theorem 3.14. *Let R be a relation scheme and let Σ be a set of FDs and MVDs which apply to R. R is in 4NF iff it is not RED_3.*

Proof.

<u>Only If</u>

As for Theorem 3.7.

<u>If</u>

The contrapositive - that if R is not in 4NF then it is RED_3 - will be established. Since R is not in 4NF, there exists a nontrivial MVD $X \rightarrow\rightarrow Y$ in Σ^+ such that X is not a superkey. If XY is not a superkey, then the conditions of Lemma 3.6 apply and the result follows immediately. Alternatively, suppose that XY is a superkey. By Lemma 3.12, the FD $X \rightarrow Z$ where $Z = R - XY$ is also in Σ^+. However, by Lemma 3.13, XZ cannot be a superkey since X is not a superkey and it follows then from Lemma 3.6 that R is RED_3. □

Next, we prove that RED_3 and RED_2 are equivalent conditions on a relation scheme.

Theorem 3.15. *A relation scheme R is RED_3 iff it is RED_2.*

Proof.

<u>If</u>

Follows immediately from the fact that $\Sigma' \subseteq \Sigma^+$.

<u>Only If</u>

We shall show the contrapositive that if R is not RED_2 then it is not RED_3. If R is not RED_2 then the attributes in every nontrivial dependency in Σ' must be a superkey or else Lemma 3.6 would imply a contradiction. Consider then any nontrivial dependency in Σ'. If the dependency is an FD $X \rightarrow Y$, then Lemma 3.13 shows that X must also be a superkey. Alternatively, if the dependency is an MVD $X \rightarrow\rightarrow Y$, then by the definition of Σ', the MVD $X \rightarrow\rightarrow Z$ where $Z = R - XY$ is also in Σ' and so both XY and XZ must be superkeys. By then using Lemma 3.12, both $X \rightarrow Y$ and $X \rightarrow Z$ must be in Σ^+ and so a simple application of the inference rules shows that X is again a superkey. Thus the left-hand side of every dependency in Σ' is a superkey and so, since $\Sigma \subseteq \Sigma'$, Σ has the same property. Then by Theorem 3.2, R is in 4NF and so, by Theorem 3.14, R is not RED_3. □

A simple consequence of the previous theorem is the following corollary which shows that a relation scheme is RED_2 with respect to one set of dependencies if and only if it is RED_2 with respect to any equivalent set of dependencies. It will be seen later that RED_1 does not possess the same desirable property.

Corollary 3.16. *Let R be a relation scheme and let Σ and Ψ be two equivalent sets of FDs and MVDs that apply to R. A relation scheme is RED_2 with respect to Σ if and only if it is RED_2 with respect with respect to Ψ.*

Proof. Immediate from the Theorem 3.15 and because equivalent sets of dependencies have the same closure. □

We now turn our attention to RED_1. It was demonstrated earlier in Example 3.1 that a relation scheme can be not RED_1 without being in 4NF and so not RED_1 is a weaker condition than 4NF. The reason for the correspondence between RED_1 and the two other types of redundancy not being valid in the case of FD and MVD constraints is that, unlike the case of FDs or MVDs alone, it is no longer true that if XY is a superkey in an FD $X \rightarrow Y$ or MVD $X \rightarrow\rightarrow Y$ then X is a superkey (see Example 3.1). The following theorem gives a syntactic characterisation of RED_1.

Theorem 3.17. *Let R be a relation scheme and let Σ be a set of FDs and MVDs that apply to R. Then the following are equivalent:*

(i) R is not RED_1;

(ii) For every nontrivial dependency $X \rightarrow\rightarrow Y$ or $X \rightarrow Y$ in Σ, XY is a superkey;

(iii) For every nontrivial dependency $X \rightarrow\rightarrow Y$ or $X \rightarrow Y$ in Σ, $X \rightarrow R - XY$ is also in Σ^+.

Proof. We shall show (i) $\Rightarrow$ (ii) $\Rightarrow$ (iii) $\Rightarrow$ (i).

(i) $\Rightarrow$ (ii)

The contrapositive, that if (ii) does not hold then R is RED_1, follows from Lemma 3.6.

(ii) $\Rightarrow$ (iii)

If $X \rightarrow\rightarrow Y$ is a nontrivial MVD in Σ such that XY is a superkey then, by Lemma 3.12, $X \rightarrow R - XY$ is also in Σ^+. Alternatively, if $X \rightarrow Y$ is a nontrivial FD in Σ such that XY is a superkey then, by Lemma 3.13, X must also be a superkey and so $X \rightarrow R - XY$ is again in Σ^+.

(iii) $\Rightarrow$ (i)

Assume to the contrary that (iii) holds but R is RED_1. Firstly, an application of inference rule A2 to (iii) shows that every dependency in Σ is a superkey. Since R is RED_1, there exists a nontrivial dependency $X \rightarrow\rightarrow Y$ or $X \rightarrow Y$ in Σ, a relation r satisfying Σ with at least two tuples identical on XY. However since r satisfies Σ, it also satisfies Σ_K which implies, by the property of a superkey, that XY cannot be a superkey which is a contradiction. □

In contrast to RED_2, the following example demonstrates that the RED_1 property has the somewhat undesirable feature that it depends on which equivalent set of dependencies is chosen.

Example 3.2. Let $R = \{A, B, C\}$, $\Sigma = \{A \rightarrow\rightarrow B, B \rightarrow A, B \rightarrow C\}$ and $\Sigma_1 = \{A \rightarrow\rightarrow C, B \rightarrow A, B \rightarrow C\}$. It follows from the inference rules that B is the only candidate key and that Σ and Σ_1 are equivalent sets of dependencies. From Theorem 3.17, R is not RED_1 with respect to Σ because every dependency contains the candidate key B. However, from Lemma 3.6, R is RED_1 with respect to Σ_1 since AC is not a superkey . □

We now address the issue of determining under what conditions the different types of redundancy are equivalent (and so conversely their absence as well). We have already seen that in the general case where the constraints contain both FD and MVD constraints, RED_1 and the two other redundancy types are not equivalent. We now show that all three types of redundancy are equivalent when the MVDs in the set of constraints are pure (see Chapter 2).

Theorem 3.18. *If R is a relation scheme and Σ is a pure set of FDs and MVDs which apply to R, then R is RED_1 if and only if it is RED_3.*

Proof.

<u>If</u>

Suppose that R is RED_3. By Theorem 3.14, if R is RED_3 then it is not in 4NF thus by Lemma 3.2 there exists a nontrivial dependency $X \rightarrow\rightarrow Y$ or $X \rightarrow Y$ in Σ with X not a superkey. XY cannot be a superkey, or else Lemma 3.12 would imply that $X \rightarrow R - XY$ was in Σ^+ thus contradicting the assumption that Σ is pure. So it follows from Lemma 3.6 that R is RED_1.

<u>Only If</u>

Immediate since $\Sigma \subseteq \Sigma^+$. □

It is noted that since RED_3 is independent of which equivalent cover is chosen by definition of Σ^+, then a simple corollary of Theorem 3.18 is that if the set of dependencies is pure, then a relation scheme is in RED_1 with respect to one set of dependencies if and only if it is in RED_1 with respect to any equivalent set of dependencies.

The property just discussed of being invariant with respect to equivalent sets of pure dependencies, but not with respect to nonpure sets, raises the question of whether some sets of dependencies are incorrectly specified. As discussed in Section 2.7, some people have proposed that certain sets of dependencies, such as conflict-free ones for instance, are more 'natural' while other people have proposed that a set of dependencies may be incomplete and extra dependencies may have to be added in order to obtain desirable properties. We would argue, based both on the previous result and intuition, that pure sets of dependencies are more natural than nonpure sets. The reason is that if a specified MVD $X \rightarrow\rightarrow Y$ is not pure then, by replacing it by either $X \rightarrow Y$ or $X \rightarrow R - XY$, an equivalent set of dependencies is obtained and so the database designer has not understood precisely the semantics of the application since an FD is a more restrictive type of constraint than an MVD [Kent 1981; Ullman 1983a].

3.7. RELATED WORK

The relationship between redundancy and normal forms has not been extensively investigated in the research literature. The only related work is that of Bernstein and Goodman, and later that by Vossen, concerning update anomalies in the case of FD constraints [Bernstein and Goodman 1980; Vossen 1988]. Although it is not immediately obvious from their definitions of update anomalies which are based on the notion of an *unpredictable update*, their definition of an update anomaly is in fact equivalent to the definitions of RED_1 and RED_3 given in this chapter. Their results are also essentially equivalent to the results derived in Section 3.4 concerning the relationship between BCNF and an absence of redundancy, though their methods of proof differ from ours. We shall discuss the relationship between normal forms and unpredictable updates in more detail in Chapter 5. In the case of MVD constraints, to our knowledge there has been no work which parallels the results derived in Sections 3.5 and 3.6.

In relation to the method for determining what set of facts is applicable to a relation, different approaches have been taken to the one adopted in this chapter where the set of facts is derived from the set of dependencies. The main alternative is to assume that the set of facts can be defined without explicit regard for the dependency constraints. This is the approach taken by Maier *et al.* [Maier et al. 1985; Maier et al. 1986; Maier and Ullman 1983; Maier et al. 1984; Maier and Warren 1982] in their studies on universal relation databases. In their approach, facts are divided into irreducible facts, called *associations*, which are essentially stored relations; and reducible facts, called *objects*, which are conceptually similar to views. They then argued that for the universal relation interface to function 'naturally', it is desirable that the set of associations and the set of objects be closed under nonempty intersection, a property that does not hold if only the attributes corresponding to an FD can be an association or object. For example, if $\Sigma = \{AB \rightarrow E, BC \rightarrow D\}$ and ABE, BCD are interpreted as associations, then this approach would require that B alone, which doesn't correspond to any FD, also be an association.

We take the view that allowing any set of attributes to be a fact can result in semantically meaningless facts. For example, given the set of attributes {*EMPLOYEE, POSITION, ADDRESS*} and the constraints $\{EMPLOYEE \rightarrow POSITION, EMPLOYEE \rightarrow ADDRESS\}$, interpreting the combination *POSITION ADDRESS* as a fact does not seem to be meaningful. The only way that a *POSITION* is associated with an *ADDRESS* is through the *EMPLOYEE*, and without it there is no semantic connection between *ADDRESS* and *POSITION*.

Another researcher, Chan [Chan 1989], has also proposed that sets of attributes which don't correspond to FDs should be allowed as facts. He argued that given a relation scheme $R = \{STUDENT, COURSE, GRADE\}$ and $\Sigma = \{STUDENT\ COURSE \rightarrow GRADE\}$, *STUDENT* and *COURSE* represent independent entities and so both should be separate facts which can be independently stored in the relation. If one allowed *STUDENT* and *COURSE* to be facts, then R would be redundant since any relation which contained at least two tuples with either the same *STUDENT* or the same *COURSE* would be redundant even though R is in BCNF.

We argue that the problem arises in this example from the assumption that all facts are contained in a single relation scheme and it disappears if multiple relation schemes are allowed. For instance, in the example of Chan, if *STUDENT* and *COURSE* are separate facts, then the situation would be best modelled by three relation schemes: R_1 = $\{STUDENT,\ STUDENT\_NAME,\ .\ .\ .\}$, $R_2 = \{COURSE,\ COURSE\_NAME,\ .\ .\ .\}$, $R_3 = \{STUDENT,\ COURSE,\ GRADE\}$. Here R_1 and R_2 record the existence of the *STUDENT* and *COURSE* entities and their properties while, as before, R_3 stores the relationship between them. In this setting, the correct way to interpret redundancy, for example in the case of *STUDENT*, is that it occurs if two tuples in R_1, rather than R_3, contain the same *STUDENT*. This is because the occurrence of the same *STUDENT* in multiple rows of R_3 doesn't represent duplicate information, it represents the distinct pieces of information that the same *STUDENT* has enrolled in multiple *COURSES*. Hence, in defining redundancy in a multiple relation setting, one should specify both the fact schemes and also the relation schemes that store them. We shall return to this issue in the final chapter.

However, although we have argued that allowing any set of attributes to be a fact is too general, the set of facts that we have defined in this chapter can be extended in a semantically meaningful way such that the results which we have derived on the relationship between the normal forms and redundancy remain valid. For example, if one views an FD $X \rightarrow A$ in an entity-relationship framework [Batini et al. 1991; Chen 1976] as meaning that "an entity named X has a property called A", then X by itself could be regarded as a fact. It is then easy to see that if the set of facts also included the sets of attributes in the left-hand sides of dependencies and the definitions of redundancy correspondingly modified, then the equivalence between 4NF, BCNF and redundancy, as given in Theorems 3.7 and 3.14, would be unchanged.

These results would also still remain valid if the set of facts was extended still further to include any candidate key. From a semantic perspective, this extension would also be plausible since there are cases where a candidate key does not appear on the left-hand side of a dependency. For instance, if in the example by Chan given previously one removed the *GRADE* attribute then, although there are no FDs in the relation scheme, *STUDENT COURSE* still makes sense as a semantic unit representing the enrolment of a *STUDENT* in a *COURSE*.

3.8. CONCLUSIONS

In this chapter, we have investigated the relationship between the absence of redundancy and the normal forms BCNF and 4NF. Based on the commonly used approach of interpreting the set of attributes in an FD or MVD constraint as the atomic unit of information, we consider redundancy to occur in a relation scheme if there exists a relation defined over the scheme with two or more tuples which are identical on the set of attributes of an FD or MVD dependency. Depending upon three different choices for the set of dependencies, we then defined three different types of redundancy in a relation scheme. For the first type, called RED_1, the set of dependencies is chosen to be Σ, the set of FDs and MVDs specified by the database design. For the second, called RED_2, the set of dependencies is chosen to be Σ plus all the MVDs which can be derived from Σ and inference rule A4 (refer to Chapter 2). For the last type, called RED_3, the set of dependencies is chosen to be all the nontrivial dependencies implied by Σ.

The major results derived in this chapter are as follows. In Section 3.3 we proved some preliminary results. The most of significant of these are the results that the 4NF, BCNF and 3NF conditions all have the desirable property that they are invariant if the set of constraints is replaced by an equivalent set of constraints. The result for 4NF is new, while the proofs for BCNF and 3NF use different techniques to those used in previous proofs. In Section 3.4, the main result derived (Theorem 3.7) is that the three types of redundancy are equivalent conditions on a relation scheme and their absence is equivalent to BCNF. In Section 3.5, the main result established is that, in the case where the only constraints are MVDs, the three types of redundancy are again equivalent conditions on a relation scheme and their absence is equivalent to 4NF. However, in Section 3.6, somewhat surprisingly, it was shown that the same result does not extend to the case of FD and MVD constraints. In this case, it was established that an absence RED_3 and RED_2 are equivalent to each other and to 4NF, but an absence of RED_1 is a weaker condition on a relation scheme. While it's not surprising that RED_1, since it does not reflect the symmetric nature of MVDs, is a weaker condition than RED_2 and RED_3, it is surprising that this weakness is only exhibited in the presence of FDs and MVDs and not in the presence of MVDs alone. A corollary of these results is that RED_2 has the desirable property that it is invariant if replaced by an equivalent set, whereas RED_1 possesses the same property in the case of FD constraints but not in the more general case of FD and MVD constraints unless the set of dependencies is pure.

CHAPTER 4

KEY-BASED UPDATE ANOMALIES AND NORMAL FORMS

4.1. INTRODUCTION

In contrast to the redundancy property analysed in the previous chapter which is a static property of a relation, in this chapter we investigate the justification for normal forms in terms of another desirable property of relations which is dynamic in nature. This property, originally introduced by Fagin, is the requirement that the integrity of a relation after an update be *easily enforced* [Fagin 1979; Fagin 1981]. Here *integrity* means that a relation satisfies the constraints and update is used in the most general sense, in other words, an update can be either the insertion of a tuple into a relation, the deletion of a tuple from a relation or the modification of an existing tuple. The work in this chapter is also reported in the literature [Vincent 1992b; Vincent and Srinivasan 1993a; Vincent and Srinivasan 1993c].

We start with a more precise definition of the phrase 'easily enforced'. In most commercial relational database software, there are no facilities in the database software for specifying a general set of constraints on a relation nor for ensuring that the integrity of a relation is maintained after an update. Instead, the only facilities for integrity enforcement in most relational systems are for the enforcement of candidate key uniqueness and, less commonly, that of ensuring that an attribute value in a tuple is in a fixed set (called a *domain constraint*). The reason for these two facilities being commonly available is that they can be simply and efficiently implemented. For instance, to enforce candidate uniqueness, two well known data structures, B-Trees or extendible hashing, are commonly used [Bayer and McCreight 1972; Fagin et al. 1979]. With extendible hashing, checking whether or not an attribute value already exists in a relation can be done in one disk access with a probability very close to one and, with a well designed B-Tree, at most two or three accesses are needed even for relations containing several hundred million tuples [El-Masri and Navathe 1989]. So to all effective purpose, checking key uniqueness can be can be done with a fixed, small number of disk accesses. Similarly, checking domain constraints can be efficiently done since in general no disk accesses are required. In contrast, checking that a general constraint, such as an FD for instance, is satisfied may require accessing all the tuples in a relation, which is clearly infeasible in most database applications.

Based on the arguments just outlined, Fagin proposed that a desirable property of a relation scheme is that the integrity of any relation defined over the scheme be guaranteed if the key constraints and domain constraints are satisfied. This formed the basis for his definition of the normal form DK/NF (refer to Chapter 3). Conversely, a *key-based update anomaly*[5] is considered as an undesirable property and is defined to occur if an update to a legal relation results in a new relation which satisfies the key and domain constraints but violates some other constraint(s). The following example illustrates the concept of an update anomaly in the case of the update being an insertion.

Example 4.1. Let $R = \{EMP,\ DEPT,\ MNGR\}$ and let $\Sigma = \{EMP \rightarrow DEPT,\ DEPT \rightarrow MNGR\}$. The only candidate key is EMP and the relation r shown in Figure 4.1 satisfies Σ. However R has an insertion anomaly since the insertion of a tuple t with the value <*Cauchy, Math, Euler*> into the relation r results in the new relation, r', satisfying the key uniqueness property but violating the FD $DEPT \rightarrow MGR$. □

[5]To avoid repetition, in this chapter the term update anomaly will refer to a key-based update anomaly.

r

EMP	DEPT	MNGR
Hilbert	Math	Gauss
Laplace	Math	Gauss
Turing	Physics	Bohr

insert <Cauchy, Math, Euler>

$\Downarrow$

r'

EMP	DEPT	MNGR
Hilbert	Math	Gauss
Laplace	Math	Gauss
Turing	Physics	Bohr
Cauchy	**Math**	**Euler**

Figure 4.1. An example of an insertion anomaly

We now outline the contribution of this chapter to the key-based approach to justifying normal forms. Fagin only considered the relationship between key-based update anomalies and normal forms for the case where the update is either an insertion or a deletion. Our major contribution is to extend this work to case of modifications of tuples. We define a *modification anomaly* as occurring when the modification of a tuple in a relation results in the violation of the constraints although both key uniqueness and a new condition - that the *identity* of the tuple be preserved by the modification - are satisfied. This additional condition is motivated by the observation that in practice it is often undesirable to change the identity of a tuple because of the need to also update associated foreign key references as well as possible confusion as to which real world entity the tuple refers to. In the relational model, a candidate key has the property of being a unique identifier and so it is natural to equate the identity of a tuple with its value on a candidate key. In general, however, a relation scheme may have several candidate keys and so there are several possibilities as to what could be interpreted as the identity of a tuple. In this thesis, three cases are considered and these are: (i) at least one (arbitrary) candidate key of the original tuple is unchanged by the modification; (ii) the primary key of the original tuple is unchanged by the modification; (iii) all candidate keys of the original tuple are unchanged by the modification. According to each of these possibilities, three different types of modification anomaly are defined (and abbreviated subsequently as MA_1, MA_2 and MA_3).

We then analyse the problem of determining necessary and sufficient conditions for the absence of these three types of modification anomalies for two classes of constraints. The first is the case where the only dependencies are FDs and the second is where the dependencies may be either FDs or MVDs. The case where the only constraints are MVDs is not considered since for this case, no modification anomalies can occur because the only candidate in this case is the relation scheme itself. For the FD case, we show that the first two types of modification anomalies, MA_1 and MA_2, are equivalent conditions on a relation scheme and a scheme is in BCNF if and only if it has neither type of anomaly. For the third type, MA_3, which leaves all candidate keys unaltered, we prove that a relation scheme is free from this anomaly if and only if it is in 3NF and the left-hand side of every FD contains only prime attributes. This last result is interesting, since the condition lies between 3NF and BCNF but is not equivalent to any of the other improvements to 3NF which have been defined in the literature [Ling et al. 1981; Smith 1978; Zaniolo 1982]. We refer to this condition as *prime attribute normal form (PANF).* We then prove that a variation of the well known synthesis [Bernstein 1976] algorithm generates relation schemes which are in PANF.

For the case where the set of dependencies includes both FDs and MVDs, we establish that, provided the set of dependencies contains at least one FD, 4NF is both a necessary and sufficient condition for a relation scheme to have no modification anomaly of type MA_1 or MA_2. However, for an MA_3, the necessary and sufficient condition is weaker than 4NF provided that the set of dependencies contains at least one pure (refer to Section 2.4.3) MVD. This new condition is that every attribute in the relation scheme is prime (i.e. a member of some candidate key - refer to Chapter 2).

The second contribution of this chapter is to strengthen one of Fagin's results [Fagin 1981] concerning normal forms and an insertion anomaly. Fagin showed that if a relation scheme is not in 4NF then there exists a relation defined over the scheme with an insertion violation (refer to the definition in Section 5.2). We prove the stronger result that, for each of the cases where the constraints contain either only FDs, or only MVDs, or only FDs and MVDs, every relation defined over a relation scheme which is not in the corresponding normal form (BCNF for FDs, 4NF for MVDs, 4NF for FDs and MVDs) has an insertion violation.

The third contribution of the chapter is to establish a new characterisation of 4NF in terms of an absence of a deletion anomaly. We prove that 4NF is equivalent to no deletion anomaly in the case where the only dependencies are MVDs and, in the case of the set of dependencies containing both FDs and MVDs, we establish the same result provided that there exists at least one pure MVD in the set of dependencies.

The sections in this chapter are organised as follows. Section 4.2 contains formal definitions and examples of all the types of key-based update anomalies. In Section 4.3, for each of the types of modification anomaly previously discussed and for an insertion anomaly, we derive necessary and sufficient conditions for a relation scheme to have no anomaly of the corresponding type for the case where the only constraints are FDs. In Section 4.4, we derive the results relating 4NF and an absence of an insertion and deletion anomaly for the case where the only constraints are MVDs. In Section 4.5 we investigate the relationship between all types of key-based update anomalies and 4NF for the most general case where the constraints contain both FDs and MVDs. Finally, Section 4.6 contains concluding remarks and comments on related work.

4.2. THE DEFINITIONS OF KEY-BASED UPDATE ANOMALIES

In this section, we present formal definitions of the different types of key-based update anomalies. The definitions of an insertion anomaly and a deletion anomaly are taken from the work of Fagin [Fagin 1981] whereas the definitions of modification anomalies are new.

4.2.1. Insertion Anomaly

Definition 4.1. Let R be a relation scheme, Σ a set of dependencies which apply to R and $r(R)$ a relation. A tuple t^* is said to be *compatible* with r if $r \cup \{t^*\}$ is a relation which is in SAT(Σ_k).

As mentioned in Chapter 2, a relation satisfies Σ_k if and only if there are no duplicate values in the relation for any candidate key, and so the compatibility condition can be equivalently characterised by the condition that $t^*[K] \notin r[K]$ for every candidate key K. We now use this concept to define an insertion violation in a relation.

Definition 4.2. A relation $r(R)$ has an *insertion violation* (*IV*) if:

(i) $r \in$ SAT(Σ);

(ii) There exists a tuple t^* defined over R such that t^* is compatible with r but $r \cup \{t^*\}$ violates Σ.

Definition 4.3. A relation scheme R has an *insertion anomaly* (*IA*) if there exists a relation $r(R)$ which has an IV.

The following example illustrates the previous definitions.

Example 4.2. Let $R = \{A, B, C\}$ and $\Sigma = \{A \rightarrow\rightarrow B, B \rightarrow A, B \rightarrow C\}$. The only candidate key is B and the relation r shown in Figure 4.2 satisfies Σ. However, R has an IA because r has an IV when the tuple $<a_1, b_2, c_2>$ is inserted into it since the resulting relation, r', satisfies the key constraints but violates the MVD $A \rightarrow\rightarrow B$. ☐

r

A	B	C
a_1	b_1	c_1

insert $<a_1, b_2, c_2>$

⇓

r'

A	B	C
a_1	b_1	c_1
$\mathbf{a_1}$	$\mathbf{b_2}$	$\mathbf{c_2}$

Figure 4.2. An example of an insertion anomaly

4.2.2. Deletion Anomaly

Definition 4.4. A relation r has a *deletion violation* (*DV*) if:

(i) $r \in \mathrm{SAT}(\Sigma)$.

(ii) There exists a tuple $t^* \in$ r such that $(r - \{t^*\}) \in \mathrm{SAT}(\Sigma_K)$ but $r - \{t^*\}$ violates Σ.

Definition 4.5. A relation scheme R has a *deletion anomaly* (*DA*) if there exists a relation $r(R)$ which has a DV.

The following example illustrates these definitions.

Example 4.3. Let $R = \{A, B, C\}$ and $\Sigma = \{A \rightarrow\rightarrow B\}$. Since Σ contains no FDs, the only candidate key is R. R has a DA because the relation r shown in Figure 4.3 has a DV when the tuple is $<a_1, b_2, c_1>$ is deleted from it since $r \in$ SAT(Σ) but the resulting relation, r', satisfies the key constraints but violates $A \rightarrow\rightarrow B$. □

r

A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_1	b_1	c_2
$\mathbf{a_1}$	$\mathbf{b_2}$	$\mathbf{c_1}$

delete $<a_1, b_2, c_1>$

$\Downarrow$

r'

A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_1	b_1	c_2

Figure 4.3. An example of a deletion anomaly

In the case of the set of constraints containing only FDs, a relation can have no deletion violation because of the well known result [Maier 1983] that if a relation satisfies a set of FDs then so does any subset of the relation.

4.2.3. Modification Anomalies

In this section, we extend Fagin's approach to the modification of tuples and define a new type of key-based update anomaly, called a modification anomaly. As mentioned previously, the motivation for defining this new type of anomaly is based on the observation that when modifying the contents of a tuple in a relation, it is often undesirable to change the value of a candidate key although, strictly speaking, this doesn't violate any of the fundamental properties of the relational model [Codd 1970]. There are several reasons for this. Firstly, from a purely relational perspective, modifying a candidate key is undesirable because of the necessity to also modify all foreign key references to the modified key. Secondly, if one interprets a candidate key as representing the identity of a tuple, then from a more general data base perspective [Khosafian and Copeland 1986], it is undesirable to have the identity of an entity change since it creates possible confusion as to what real world entity the data actually refers to. Codd supported this approach in his later extensions to the relational model [Codd 1979] where he proposed the use of internally generated, unchangeable identifiers which he referred to as *surrogates*. Also, most of the new generation data base models, such as object-oriented data models [Abiteboul and Kanellakis 1989; Kim 1990], support immutable object identity.

Essentially, we define a key-based modification anomaly as occurring when the modification of a tuple doesn't change the identity of a tuple or violate key uniqueness, but the resulting relation still violates the FD and MVD constraints. However, unlike many object-oriented data models which support only a single object identifier, in the relational model there may be multiple candidate keys for a relation scheme and so there are several possible interpretations as to what is meant by leaving the identity of a tuple unchanged. In increasing restrictiveness, they are:

(i) the replacement tuple is identical to the original on *at least one (arbitrary) candidate key*;

(ii) the replacement tuple is identical to the original on *the primary key*;

(iii) the replacement tuple is identical to the original on *every candidate key*.

For each of these alternatives, we now present a formal definition of a modification anomaly by modelling a modification as a deletion followed by an insertion.

Definition 4.6. A relation $r(R)$ has a *modification violation 1* (MV_1) with respect to a set Σ of FDs and MVDs if there exists a tuple $t \in r$ and a tuple t^* defined over R such that:

(i) $r \in \mathrm{SAT}(\Sigma)$;

(ii) t^* is compatible with $(r - \{t\})$;

(iii) t and t^* are identical on *at least one (arbitrary) candidate key*;

(iv) $(r - \{t\}) \cup \{t^*\}$ violates Σ.

Definition 4.7. A relation $r(R)$ has a *modification violation 2* (MV_2) with respect to a set Σ of FDs and MVDs if it satisfies all conditions of Definition 4.6 except that condition (iii) is changed to:

(iii') t and t^* are identical on the *primary key*.

Definition 4.8. A relation $r(R)$ has a *modification violation 3* (MV_3) with respect to a set Σ of FDs and MVDs if it satisfies all the conditions of Definition 4.6 except that condition (iii) is changed to:

(iii'') t and t^* are identical on *every candidate key*.

We now use these definitions of violations in relation instances to define anomalies in the corresponding relation schemes.

Definition 4.9. A relation scheme R has a *modification anomaly 1* (MA_1) if there exists a relation $r(R)$ which has an MV_1.

Definition 4.10. A relation scheme R has a *modification anomaly 2* (MA_2) if there exists a relation $r(R)$ such that r has an MV_2.

Definition 4.11. A relation scheme R has a *modification anomaly 3* (MA_3) if there exists a relation $r(R)$ such that r has an MV_3.

From these definitions, it is easily seen that the following implications hold: r has an $MV_3 \Rightarrow r$ has an $MV_2 \Rightarrow r$ has an MV_1; and hence the following implications also hold for relation schemes: $MA_3 \Rightarrow MA_2 \Rightarrow MA_1$.

The following example illustrates the previous definitions.

Example 4.4. Consider the case of $R = \{A, B, C, D\}$ and $\Sigma = \{ABC \rightarrow D, D \rightarrow C, B \rightarrow\rightarrow A\}$. It can be verified that the candidate keys are ABC and ABD. The relation r shown in Figure 4.4 is in SAT(Σ).

If the tuple $t = \langle a_2, b_1, c_1, d_1 \rangle$ is changed to $t^* = \langle a_2, b_1, c_1, \boldsymbol{d_2} \rangle$, resulting in the relation r' shown in Figure 4.4, then r has an MV_1. To verify this, each of the conditions of an MV_1 (Definition 4.6) will be shown to hold. Condition (i) is satisfied since, as mentioned earlier, r is in SAT(Σ). Condition (ii) follows from the fact that both tuples in r' are distinct on both candidate keys. Condition (iii) holds since t and t^* are identical on the candidate key ABC and condition (iv) is satisfied because r' violates $B \rightarrow\rightarrow A$.

If ABC is chosen as the primary key, then r also has an MV_2 when t is replaced by t^* since $B \rightarrow\rightarrow A$ is still violated. If ABD is chosen as the primary key, then replacing t by t^* does not constitute an MV_2 since t and t^* are not identical on the primary key.

However, if instead t is replaced by $<a_2, b_1, \mathbf{c}_2, d_1>$, resulting in the relation r'' shown in Figure 4.4, then r has an MV_2 since $r'' \in$ SAT(Σ_K) but $r'' \notin$ SAT(Σ) because it violates $B \rightarrow\rightarrow A$.

It is interesting to note however that neither r nor any other relation defined over R can have an MV_3. This follows because every attribute in R is prime and so any modified tuple satisfying condition (iii") in Definition 4.8 must be identical to the original, but then conditions (i) and (iv) cannot be satisfied simultaneously. □

r

A	B	C	D
a_1	b_1	c_1	d_1
a_2	b_1	c_1	d_1

r'

A	B	C	D
a_1	b_1	c_1	d_1
a_2	b_1	c_1	$\mathbf{d}_2$

r"

A	B	C	D
a_1	b_1	c_1	d_1
a_2	b_1	$\mathbf{c}_2$	d_1

Figure 4.4. An example illustrating modification violations

4.3. THE CASE OF FD CONSTRAINTS

In this section we derive several of the main results of this chapter concerning the necessary and sufficient conditions for an absence of key-based update anomalies in the case where the only constraints are FDs.

4.3.1. Insertion Anomaly and Normal Forms

In this section we derive a result which strengthens a result obtained by Fagin [Fagin 1981] concerning the relationship between BCNF and the absence of an insertion anomaly in a relation scheme for the case of FD constraints. Fagin showed that if a relation scheme R is not in BCNF, then there exists at least one legal relation defined over R which has an insertion violation. We establish the stronger result that every legal relation defined over a relation scheme that is not in BCNF has an insertion violation. Our method of proof differs from that of Fagin's since ours is based on direct construction, whereas his method is an indirect one based on the characterisation of BCNF in terms of the satisfaction of key constraints (see Section 3.3). A simple consequence of this result is that a relation scheme is in BCNF if and only if it has no insertion anomaly. Firstly, a preliminary lemma is established.

Lemma 4.1. *If a set of attributes X is not a superkey, then for every candidate key K, $K - X = \emptyset$.*

Proof. If $K \subseteq X$, then a simple application of the inference rules derives the contradiction that X is a superkey. □

Theorem 4.2. *Let R be a relation scheme and Σ a set of FDs which apply to R. If R is not in BCNF then every nonempty relation $r(R) \in SAT(\Sigma)$ has an IV.*

Proof. If R is not in BCNF then there exists a nontrivial FD $X \rightarrow Y \in \Sigma^+$ where X is not a superkey. Let t be any tuple in r and let t^* be the tuple defined by $t^*[X] = t[X]$ and $t^*[A] \notin r[A]$ for all other attributes $A \in (R - X)$. Such a tuple exists because a relation contains only a finite number of tuples but attribute domains are assumed to contain an infinite number of values. The claim is that r has an insertion violation when t^* is inserted into it.

To verify this claim, one firstly has to show that t^* is compatible with r. It follows from Lemma 4.1 that for every candidate key K, $K - X = \emptyset$ and so by definition of t^*, $t^*[K] \notin r[K]$ and thus the compatibility condition is satisfied. Next, we show that $r \cup \{t^*\}$ violates Σ. Since $X \to Y$ is nontrivial, $Y - X = \emptyset$. So by construction of t^*, $t^*[Y] \neq t[Y]$ and so $X \to Y$ is violated in $r \cup \{t^*\}$. □

The main result of this section is the following theorem.

Theorem 4.3. *A relation scheme R is in BCNF iff it has no IA.*

Only If

Let R be in BCNF and suppose to the contrary that R has an insertion anomaly where a nontrivial FD $X \to Y$ is violated. For this to occur, there has to be a relation r and at least two distinct tuples t_1 and t_2 in r such that $t_1[X] = t_2[X]$. But by definition of IA, $r \in \mathrm{SAT}(\Sigma_K)$ and so X cannot be a superkey which contradicts the assumption that R is in BCNF.

If

The contrapositive follows from Theorem 4.2 and the fact that, since any relation which contains no duplicate values for any attribute satisfies Σ, there is an infinite number of relations which satisfy Σ since attribute domains are assumed to be infinite. □

It is assumed that the domains for attributes are infinite in the above result. If this assumption is dropped, then Theorem 4.2 is no longer valid since the proof requires that we can always choose an attribute value which is not in a relation. However, provided that each domain contains at least two values, Theorem 4.2 (and thus Theorem 4.3 as well) will still be valid provided that Theorem 4.2 is replaced by a weaker statement that every relation containing one tuple has an IV.

4.3.2. MA_1 Anomaly and Normal Forms

In this section, we investigate the necessary and sufficient conditions for a relation scheme to have no MA_1. The main result that we derive (Theorem 4.7) is that BCNF is equivalent to no MA_1 in a relation scheme. Some preliminary lemmas are derived first.

Lemma 4.4. *Let Σ be a set of FDs and let $X \rightarrow A$ be an FD in Σ. If X is not a superkey then XA is not a superkey.*

Proof. By the simple application of the inference rules. □

Lemma 4.5. *Let $r(R)$ be a relation in $SAT(\Sigma)$ and let X be a set of attributes. If X is not a superkey, then for any tuple $t \in r$ there exists a tuple t' such that $t[X] = t'[X]$ and the relation $r \cup \{t'\} \in SAT(\Sigma)$.*

Proof. The construction and proof is similar to that used in establishing the completeness of the FD inference rules [Armstrong 1974; Fagin 1977b; Maier 1983]. Define the tuple t' as follows. For every attribute $A \in R$, if $A \in X^+$ then let $t'[A] = t[A]$ otherwise set $t'[A]$ to a value such that $t'[A] \notin r[A]$. This can be done since attribute domains are assumed to be infinite. The claim is that t' satisfies the conditions of the theorem. Firstly, it satisfies $t[X] = t'[X]$ because, from inference rule A1, $X \subseteq X^+$. Secondly, since X is not a superkey, $X^+ \neq R$ and so there has to be at least one attribute $B \in (R - X^+)$. But by construction of t', $t'[B] \notin r[B]$ and so $r \cup \{t'\}$ is actually a relation. Next it has to be verified that $r \cup \{t'\} \in$ SAT(Σ). Let $Z \rightarrow Y$ be any FD in Σ. If $Z - X^+ = \emptyset$, then by the inference rule A3, Y must also be a subset of X^+ and so by the construction of t', $t'[ZY] = $ t$[ZY]$. This implies that $r \cup \{t'\}$ must satisfy $Z \rightarrow Y$ since

$r \in$ SAT(Σ). Alternatively, if $Z - X^+ \neq \emptyset$ then by construction of t', $t'[Z] \notin r[Z]$ and so $Z \rightarrow Y$ is again satisfied. □

We now use these preliminary lemmas to show that for every nonempty relation defined on a scheme which is not in BCNF, one can always add a tuple so that the resulting relation has an MV_1.

Lemma 4.6. *Let R be a relation scheme and let Σ be a set of FDs. If R is not in BCNF, then for every nonempty relation $r(R) \in SAT(\Sigma)$ there exists a tuple t' such that $r \cup \{t'\}$ has an MV_1.*

Proof. Since R is not in BCNF, there exists a nontrivial FD $X \rightarrow A$ in Σ^+ such that X is not a superkey. Let r be any relation in SAT(Σ). Since X is not a superkey, then by Lemma 4.4 XA is not a superkey and so by Lemma 4.5 there exists a tuple $t \in r$ and a tuple t' such that $t[XA] = t'[XA]$ and $r \cup \{t'\} \in$ SAT(Σ). Let t^* be the tuple obtained by changing the value of $t'[A]$ to some other value such that $t^*[A] \notin r[A]$ and modify r by replacing t' with t^*. The claim is that $r \cup \{t'\}$ has an MV_1 when t' is replaced by t^*. Each of the conditions of an MV_1 (Definition 4.6) will be verified in turn. Condition (i) follows from Lemma 4.5. Next, if a candidate key K doesn't contain A, then by definition of t^*, $t^*[K] = t'[K]$ and so t^* is compatible with $(r - \{t'\})$ because t' is compatible with r since $r \cup \{t'\} \in$ SAT(Σ). If K contains A, then $t^*[K] \notin r[K]$ by construction of t^* and so (ii) is again satisfied. Since $X \rightarrow A$ is nontrivial, applying the inference rules shows that $R - A$ is a superkey and so there must exist at least one candidate key $K' \subseteq (R - A)$. Then since t^* and t' only differ on A, it follows that $t^*[K'] = t'[K']$ and (iii) is satisfied. Lastly, again since $X \rightarrow A$ is nontrivial, it follows by the construction of t^* that $X \rightarrow A$ is violated and so (iv) is satisfied. □

We note that this lemma cannot be strengthened to show that any nonempty relation defined over a scheme which is not in BCNF must have an MV_1. This is demonstrated in the following well known example.

Example 4.5. Let $R = \{A, B, C\}$, $\Sigma = \{AB \rightarrow C, C \rightarrow B\}$ and $r = \{t_1, t_2\}$ where $t_1 = \langle a_1, b_1, c_1 \rangle$ and $t_2 = \langle a_2, b_1, c_2 \rangle$. The relation r is shown in Figure 4.5. Since the candidate keys are AB and AC, R is not in BCNF because of the FD $C \rightarrow B$. We shall show that r has no MV_1. Consider the modification of either t_1 or t_2. Any modification which leaves both candidate keys unchanged cannot result in an MV_1 since every attribute in R is prime. Suppose firstly that t_1 is modified to t' so that, as shown by the relation r' in Figure 4.5, $t'[AB] = t_1[AB]$. Then $AB \rightarrow C$ is still satisfied since $t'[AB] \neq t_2[AB]$ and $C \rightarrow B$ is still satisfied since $t'[B] = t_2[B]$. Alternatively, if t_1 is modified to t' such that, as shown by the relation r'' in Figure 4.5, $t'[AC] = t_1[AC]$ then $AB \rightarrow C$ is still satisfied because $t'[A] \neq t_2[A]$ and $C \rightarrow B$ is satisfied because $t'[C] \neq t_2[C]$. Similar arguments apply if t_2 is modified and so r does not have an MV_1. □

r

A	B	C
a_1	b_1	c_1
a_2	b_1	c_2

r'

A	B	C
a_1	b_1	$\mathbf{c_3}$
a_2	b_1	c_2

r"

A	B	C
a_1	$\mathbf{b_2}$	c_1
a_2	b_1	c_2

Figure 4.5. An example of a relation with no MV_1

These lemmas lead to the main theorem of this section.

Theorem 4.7. *A relation scheme is in BCNF if and only if it has no MA_1.*

Proof.

Only If

As for Theorem 4.3.

If

The contrapositive, that if R is not in BCNF then it has an MA_1, follows directly from Lemma 4.6 and the fact that there is an infinite number of relations in SAT(Σ), since attribute domains are infinite and any relation which contains no duplicate values for any attribute satisfies Σ. □

4.3.3. MA_2 Anomaly and Normal Forms

We investigate in this section the necessary and sufficient conditions for a relation scheme to have no MA_2. The main result that we derive (Theorem 4.12) is that BCNF is equivalent to no MA_2 in a relation scheme. Firstly, we present some preliminary lemmas.

Lemma 4.8. *Let Σ be a reduced set of FDs. If, for any FD $X \rightarrow A \in \Sigma$, X is a superkey then it must also be a candidate key.*

Proof. If X is a superkey but not a candidate key then there must exist a proper subset X' of X such that X' is a candidate key. But then a simple application of the inference rules shows that Σ - $\{X \rightarrow A\} \cup \{X' \rightarrow A\} \equiv \Sigma$ which contradicts the assumption that Σ is reduced. □

Lemma 4.9. *If K is a candidate key then there cannot exist a nontrivial FD $X \rightarrow A$ in Σ^+ such that $XA \subseteq K$.*

Proof. Assume to the contrary that there is such an FD. Let $K' = K - A$. Then since $X \rightarrow A$ is nontrivial, $A \notin X$ and so $X \subseteq K'$ and thus $K' \rightarrow A$ follows from rules A1 and A3. But using rule A2 and augmenting both sides of $K' \rightarrow A$ by K', this implies that $K' \rightarrow K$ and hence $K' \rightarrow R$ from rule A3 and the fact that K is a candidate key. This is a contradiction since by definition a candidate key cannot have a proper subset which is a superkey. □

Lemma 4.10. *If Σ is a set of FDs and $X \rightarrow A$ an FD in Σ such that X is not a superkey, then for every candidate key K, $K - XA \neq \emptyset$.*

Proof. Suppose that there is a candidate key K such that $K \subseteq XA$. From Lemma 4.4, since X is not a superkey, XA cannot be a superkey, but since K is a candidate key applying rules A1 and A3 shows that $XA \rightarrow R$ contradicting the fact that XA cannot be a superkey. □

The next lemma is needed for the construction of counter-examples in the proofs of the main theorems concerning MA_2 and MA_3.

Lemma 4.11. *Let Σ be a reduced set of FDs and let $X \rightarrow A$ be an FD in Σ such that X is not a superkey. Also, let V and Y be two subsets of X such that $X = V \cup Y$, $V \cap Y = \emptyset$, $V \neq \emptyset$, $Y \neq \emptyset$. Construct a relation r of two tuples, t_1 and t_2, such that t_1 and t_2 are identical on V^+ and different elsewhere. Then r has the following properties:*

(i) $r \in SAT(\Sigma)$;

(ii) $t_1[V] = t_2[V]$;

(iii) $t_1[Y] \neq t_2[Y]$;

(iv) $t_1[A] \neq t_2[A]$.

Proof. Properties (i) and (ii) follow from the fact that V is not a superkey since X is not a superkey and a well known result on two tuple relations (Theorem 4.1 in [Maier 1983]). To establish (iii), assume to the contrary that $t_1[Y] = t_2[Y]$. Then by the definition of r, $V \rightarrow Y$ must be in Σ^+ and a straightforward application of the inference rules shows that $V \rightarrow A$ is in Σ^+ which contradicts the assumption that Σ is reduced since V is a proper subset of X. Similarly, (iv) holds since otherwise the reduced assumption would be violated by again replacing $X \rightarrow A$ in Σ by $V \rightarrow A$. □

These results are now used to establish the main theorem of this section.

Theorem 4.12. *If R is a relation scheme and Σ a set of FDs which apply to R, then R is in BCNF if and only if it has no MA_2.*

Proof.

<u>Only If</u>

As for Theorem 4.3.

<u>If</u>

The contrapositive, that if R is not in BCNF then it has an MA_2, will be established. If R is not in BCNF then there exists a nontrivial FD $X \rightarrow A$ in Σ^+ such that X is not a superkey. Then, from Corollary 4.3, there has to be a nontrivial FD in Σ itself such that the left-hand side of the FD is not a superkey. However, by the construction shown earlier in Chapter 2, every set of FDs has a reduced cover and so Σ has a BCNF violation if and only if a reduced cover for Σ also has a BCNF violation. So without loss of generality Σ can be assumed to be a reduced set and $X \rightarrow A$ an FD in Σ which violates BCNF. The proof is divided into three separate cases which cover every possibility. These cases are:

(i) $A \notin K$ where K is the primary key of R;

(ii) $A \in K$ and $X \cap K = \emptyset$;

(iii) $A \in K$ and $X \cap K \neq \emptyset$.

(i) $A \notin K$

Since by assumption X is not a superkey then, by Lemma 4.4, XA is not a superkey. Let r be a relation of one tuple t whose attribute values are chosen arbitrarily from each of the domains. Then r automatically is in SAT(Σ), so by Theorem 4.2 there exists a tuple t' such that $t'[XA] = t[XA]$ and $r \cup \{t'\} \in$ SAT(Σ). Let t^* be the tuple obtained by modifying $t'[A]$ to a value distinct from $t[A]$, i.e. $t^*[A] \neq t[A]$. It can be easily verified that r has an MV_2 when t' is updated to t^*.

(ii) $A \in K$ and $X \cap K = \emptyset$

In this case, construct a relation r of two tuples t_1, t_2 such that for every attribute $B \in R$, $t_1[B] \neq t_2[B]$. Obviously $r \in$ SAT(Σ). Let t^* be the tuple such that $t^*[X] = t_1[X]$ and $t^*[R - X] = t_2[R - X]$. The claim is that r has an MV_2 when t_2 is updated to t^*. To verify this, condition (i) of MV_2 follows by construction of t_1 and t_2.

To verify (ii), since X is not a superkey then, by Lemma 4.1, for every candidate key K', $K' - X \neq \emptyset$. Hence by the construction of t^*, $t^*[K'-X] = t_2[K'-X]$ and so $t^*[K'] \neq t_1[K']$ by construction of t_1 and t_2. Condition (iii') follows by the construction of t^* and the fact that $X \cap K = \emptyset$. Condition (iv) follows by construction of t^* and the fact that $t^*[A] \neq t_1[A]$ since $X \rightarrow A$ is nontrivial.

(iii) $A \in K$ and $X \cap K \neq \emptyset$

Let $V = X \cap K$ and $Y = X - K$. It follows directly from these definitions that $X = V \cup Y$, $V \cap Y = \emptyset$. Also, $V \neq \emptyset$ by assumption and $Y \neq \emptyset$ since otherwise it would follow that $XA \subseteq K$ since $A \in K$, and so from Lemma 4.9 one would derive the contradiction that K is not a candidate key. Thus the conditions of Lemma 4.11 are satisfied and the two tuple relation r for which the tuples are identical on V^+ and different elsewhere has the properties given by the lemma. Let t^* be the tuple defined by $t^*[Y] = t_1[Y]$ and $t^*[R - Y] = t_2[R - Y]$. The claim is that r has an MV_2 when t_2 is replaced by t^*. Each of the conditions of an MV_2 will now be verified in turn.

Condition (i) is automatically satisfied by (i) of Lemma 4.11. By Lemma 4.9, for any candidate key K', $K' - VYA \neq \emptyset$. Also, $K' - V^+ \neq \emptyset$ or else a simple application of the inference rules would imply the contradiction that X was a superkey, and combining this with the previous result gives $K' - V^+YA \neq \emptyset$. Thus by definition of r, $t_1[K' - V^+YA] \neq t_2[K' - V^+YA]$ and so by definition of t^*, it follows that $t^*[K'] \neq t_1[K']$ and the compatibility condition (ii) is satisfied. Condition (iii') follows since by definition of t^*, t^* and t_2 differ only on attributes in $X - K$. Condition (iv) follows from Lemma 4.11 and the construction of t^*. □

It is also noted that since MA_2 implies MA_1, then a corollary of the previous theorem is an alternative proof of Theorem 4.7.

4.3.4. MA_3 Anomaly and Normal Forms

The relationship between an MA_3 anomaly and the syntactic normal forms BCNF and 3NF is investigated in this section. We show that BCNF is a stronger condition than is needed to avoid an MA_3, and the necessary and sufficient is a new normal form, called PANF, which lies in between 3NF and BCNF. Firstly, as shown in the following lemma, it is straightforward to verify that BCNF is a sufficient condition for a relation scheme to have no MA_3.

Lemma 4.13. *If R is a relation scheme and Σ a set of FDs which apply to R, then if R is in BCNF it has no MA_3.*

Proof. As for Theorem 4.3. □

Unlike the case for MA_1 and MA_2, the converse to Lemma 4.13 doesn't hold. As the following example demonstrates, a relation scheme may have no MA_3 yet not be in BCNF.

Example 4.6. Let $R = \{STUDENT,\ COURSE,\ TEACHER\}$ and let $\Sigma = \{STUDENT\ COURSE \rightarrow TEACHER,\ TEACHER \rightarrow COURSE\}$. Then both STUDENT COURSE and STUDENT TEACHER are candidate keys and so every attribute in *R* is prime. No relation defined over *R* can have an MV_3. This follows because every attribute in *R* is prime and so any modified tuple satisfying condition (iii") in Definition 4.8 must be identical to the original, but then conditions (i) and (iv) cannot be satisfied simultaneously. □

In the following theorem, a necessary and sufficient condition for a relation scheme to have no MA_3 is derived.

Theorem 4.14. *Let R be a relation scheme and let Σ be a reduced set of FDs which apply to R. R has no MA_3 iff:*

(i) R is in 3NF;

(ii) For every FD $X \to A$ in Σ, X contains only prime attributes.

Proof.

<u>If</u>

Suppose to the contrary that (i) and (ii) are satisfied but R has an MA_3. Then there exits a relation r and a tuple $t \in r$ such that at least one FD $X \to A$ is violated when t is modified to t^*. For this to happen, there has to exist a tuple $t' \in r$ and an FD $X \to A$ in Σ such that $t'[X] = t^*[X]$ and $t'[A] \neq t^*[A]$. Because of (ii) and the definition of an MA_3, $t^*[X] = t[X]$, and since $r \in \text{SAT}(\Sigma)$, $t[A] = t'[A]$ and so $t^*[A] \neq t[A]$. However, because $t^*[X] = t'[X]$ and the modified relation is in $\text{SAT}(\Sigma_K)$ by definition of an MV_3, X cannot be a superkey and so by (i) this implies that A must be a prime attribute. Hence by definition of an MV_3, this implies that $t^*[A] = t[A]$ thus contradicting the fact that $t^*[A] \neq t[A]$.

<u>Only If</u>

(i)

The contrapositive that if R is not in 3NF then it has an MA_3 will be established.

If R is not in 3NF then there is an FD $X \to A$ in Σ^+ such that X is not a superkey and A is not a prime attribute. Then the same construction used in part (i) of the proof of Theorem 4.12 shows that R has an MA_3.

(ii)

As before, we shall establish the contrapositive that if there exists an FD $X \to A$ in Σ such that X contains nonprime attributes then R has an MA_3. Suppose firstly that X

contains only nonprime attributes. The same construction used in part (ii) of Theorem 4.12 shows that R has an MA_3.

Alternatively, suppose that X contains both prime and nonprime attributes. Let V be the set of prime attributes in X and let Y be the set of nonprime attributes in X. X cannot be a superkey or else by Lemma 4.8 it would be a candidate key thus contradicting the assumption that it contains nonprime attributes. Then the conditions of Lemma 4.11 are satisfied and the same construction to that used in (iii) of Theorem 4.12 shows that r has an MV_3. □

As will be shown by the following examples, conditions (i) and (ii) in Theorem 4.14 are not comparable. There exist relation schemes that satisfy one condition but not the other.

Example 4.7. Let $R = \{A, B, C\}$ and $\Sigma = \{B \rightarrow C\}$. The only candidate key is AB and so R satisfies (ii) since B is prime but not (i) since B is not a superkey and C is not prime. □

Example 4.8. Let $R = \{A, B, C, D\}$ and let $\Sigma = \{AB \rightarrow C, CD \rightarrow AB, BD \rightarrow A\}$. The candidate keys are CD and BD and so R is in 3NF since C is a prime attribute, but R is not in PANF since A is not a prime attribute. □

Any relation scheme satisfying the conditions in Theorem 4.14 will be said to be in *prime attribute normal form* (*PANF*).

4.3.4. Comparing PANF With Other Normal Forms

Two other normal forms which are improvements of 3NF, namely *elementary key normal form* (*EKNF*) [Zaniolo 1982] and *improved 3NF* [Ling et al. 1981], have also been defined and in this section we compare them to PANF. Firstly, improved 3NF only applies to the case of multiple relations and since we are only concerned with single relations in this work we don't consider it any further. As for EKNF, which was formally defined earlier in Chapter 3, we now construct examples to show that EKNF and PANF are not comparable, i.e. there are schemes which are in PANF but not in EKNF, and conversely there are schemes which are in EKNF but not in PANF. The first example, taken from Zaniolo's work [Zaniolo 1982], shows that a scheme can be in PANF but not in EKNF, while the second example demonstrates the converse.

Example 4.9. Let $R = \{D\#, MGID, ACC\#\}$ and $\Sigma = \{D\# \rightarrow MGID, MGID \rightarrow D\#\}$. Then the candidate keys are $D\#\ ACC\#$ and $MGID\ ACC\#$. Neither of these candidate keys is elementary because neither of the FDs $D\#\ ACC\# \rightarrow MGID$ nor $MGID\ ACC\# \rightarrow D\#$ is an elementary FD and so there are no elementary key attributes. Hence R is not in EKNF since if one considers the FD $D\# \rightarrow MGID$ then $D\#$ is not a candidate key nor is $MGID$ an elementary key attribute. However, R is in PANF since every attribute in R is prime. □

Example 4.10. As in Example 4.8, let $R = \{A, B, C, D\}$ and $\Sigma = \{AB \rightarrow C, CD \rightarrow AB, BD \rightarrow A\}$. Both the candidate keys BD and CD are elementary because of the FDs $CD \rightarrow AB$, $BD \rightarrow A$ and so B, C and D are elementary key attributes. Thus R is EKNF since both CD and BC are superkeys and C is an elementary key attribute. However, we showed in Example 4.8 that R is not in PANF because A is not prime. □

4.3.5. Achieving PANF

In this section we consider the problem of generating relation schemes which are in PANF. We start with the simplest version [Ullman 1988a] of the well known synthesis algorithm due to Bernstein [Bernstein 1976] for generating 3NF schemes.

ALGORITHM 4.1. A synthesis algorithm for achieving 3NF.

Input: A relational scheme R and a reduced set Σ of FDs which apply to it.

Output: A dependency preserving, lossless decomposition of R into 3NF.

Method:

For each FD $X \rightarrow A$ in Σ, create the scheme XA. If there is no scheme which contains a candidate key K then create an extra scheme which contains K alone.

It can be shown [Biskup et al. 1979] that the relation schemes generated by this algorithm are in 3NF and the decomposition is dependency preserving and lossless [Aho et al. 1979a]. There are several differences between Algorithm 4.1 and Bernstein's original synthesis algorithm. Firstly, Algorithm 4.1 adds a candidate key, if necessary, as one of the relation schemes. This enhancement was proposed [Biskup et al. 1979] after the publication of the original synthesis algorithm and it ensures that the decomposition is lossless, a property that the original synthesis algorithm did not possess. Secondly, in order to minimise the number of relation schemes, the original synthesis algorithm extends Algorithm 4.1 by merging relation schemes which are generated from FDs with equivalent left-hand sides. In other words, if two schemes XA and YB corresponding to two reduced FDs $X \rightarrow A$ and $Y \rightarrow A$ are generated from Algorithm 4.1 (X and Y may be the same) and both $X \rightarrow Y$ and $Y \rightarrow X$ are in Σ^+, then XA and YB are merged into a single scheme $XYAB$. A complication that then arises is that the merged scheme may not then be in 3NF and extra steps are required to remove 3NF violations.

We now show that the relation schemes generated by Algorithm 4.1 are in PANF.

Theorem 4.15. *Each of the relation schemes generated by Algorithm 4.1 is in PANF.*

Proof. If the relation scheme is a candidate key then the result is immediate, so alternatively assume that it is the scheme $R' = XA$ which corresponds to the FD $X \rightarrow A$ in Σ. We note from the properties of projected FDs [Maier 1983; Ullman 1988a] that any FD which holds in R' must also hold in R. We show firstly that X must be a candidate key in R'. X is a superkey in R' because $R' = XA$. It is also a candidate key since if not there must exist a candidate key $K \in R'$ with $K \subset X$, and thus $K \rightarrow A \in \Sigma^+$ which contradicts the assumption that $X \rightarrow A$ is reduced. Let $Y \rightarrow B$ be any FD which holds in R'. We divide the proof into the two cases $B = A$ and $B \neq A$.

<u>B= A</u>

For this case, one must have that $Y = X$ since the only other possibility is for $Y \subset X$ and this would violate the property that $X \rightarrow A$ is reduced for the reasons outlined previously. We have already shown that X is a candidate key in R' and so the result follows.

<u>B ≠ A.</u>

Firstly, since $R' = XA$ then $B \in X$ and so, from Lemma 4.9, Y cannot also be a subset of X and thus Y can be written as AX' where $X' \subset X$. X' is prime in R' since X is prime in R' and so it remains to show that A is prime in R'. Assume to the contrary that it is not. Since $AX' \rightarrow B$ and $R' = AX'B$, AX' - B is a superkey in R' and so there exists a candidate key K such that $K \subseteq AX'$ - B. Then since A is not prime in R', $K \subseteq X$ - B and so $K \subset X$ which contradicts the fact that $X \rightarrow A$ is reduced by the same argument as before. □

4.4. THE CASE OF MVD CONSTRAINTS

In this section, we derive results concerning the relationship between syntactic normal forms and update anomalies for the case where the set of constraints contains only MVDs. We don't consider modification anomalies in this section because, as mentioned in the introduction to this chapter, modification anomalies don't exist in this case. This is because no FDs are implied by a set of MVDs[6] and so the only candidate key in a relation scheme is the scheme itself, thus it follows directly from the definitions of the modification anomalies that a relation scheme can have no type of modification anomaly.

4.4.1. Insertion Anomaly and Normal Forms

The main results derived in this section are to show that the results of Section 4.3.1 extend to the MVD case.

Theorem 4.16. *Let R be a relation scheme and Σ a set of MVDs which apply to R. If R is not in 4NF then every nonempty relation r(R) which is in SAT(Σ) has an IV.*

Proof. If R is not in 4NF then there exists a nontrivial MVD $X \rightarrow\rightarrow Y \in \Sigma^+$ where X is not a superkey. As per Theorem 4.2, let t be any tuple in r and let t^* be the tuple defined by $t^*[X] = t[X]$ and $t^*[A] \notin r[A]$ for all other attributes $A \in (R - X)$. The claim is that r has an insertion violation when t^* is inserted into it.

To verify this claim, the compatibility condition is satisfied because the only candidate key is R and $X \rightarrow\rightarrow Y$ is violated in $r \cup \{t^*\}$ since, by the definition of t^* and the fact that $X \rightarrow\rightarrow Y$ is nontrivial, $t^*[X] = t[X]$ and $t^*[Y] \notin r[Y]$ and $t^*[R - XY] \notin r[R - XY]$. □

[6] Corollary to Theorem 8.11 in Maier [Maier 1983].

The above theorem provides the following characterisation of 4NF in terms of an absence of an insertion anomaly.

Theorem 4.17. *A relation scheme R is in 4NF iff it has no IA.*

Only If

As for Theorem 4.3.

If

The contrapositive follows from Theorem 4.16 and the fact that, since any relation which contains no duplicate values for any attribute satisfies Σ, there is an infinite number of relations which satisfy Σ. □

4.4.2. Deletion Anomaly and Normal Forms

In the case of a deletion anomaly, the analogue of Theorem 4.16 does not hold since any relation which satisfies a set of MVDs as FDs can have no deletion anomaly because no legal relation can have a DV when the only dependencies are FDs. However, although not every relation has a deletion violation, the following analogue of Theorem 4.17 shows that every non 4NF relation scheme has at least one relation defined over it which has a deletion violation.

Theorem 4.18. *A relation scheme R is in 4NF iff it has no DA.*

Proof.

Only If

As for Theorem 4.3.

If

We shall show the contrapositive that if R is not in 4NF then it has a DA. If R is not in 4NF then there must be a nontrivial MVD $X \rightarrow\rightarrow Y \in \Sigma^+$ with X not a superkey. Since the MVD is nontrivial, $R - XY \neq \emptyset$. Construct then the tableau T_X as shown in Figure 4.6 where x, y and z indicate vectors of distinguished variables and y' and z' indicate vectors of nondistinguished variables. Then the J-rule for $X \rightarrow\rightarrow Y$ can be applied twice to yield the tableau T' also shown in Figure 4.6. Since only MVDs are being considered and the corresponding J-rule only adds rows to a tableau, and the result of the chase is independent of the sequence in which the rules are applied, then T^*, where $T^* = chase_\Sigma(T_X)$, must contain the rows in T'. Let ρ be any one-to-one valuation for T^*. The claim is that ρ(T^*) has a deletion violation.

Firstly, since the rows in T' are distinct and ρ is one-to-one, it follows that the corresponding tuples in ρ(T^*) are also distinct. If any of the tuples corresponding to the rows of T' are deleted from ρ(T^*), it follows directly from the definition of an MVD that the new relation violates $X \rightarrow\rightarrow Y$ and so ρ(T^*) has a deletion violation. □

T_X

X	Y	R - XY
x	y	z
x	y'	z'

T'

X	Y	R - XY
x	y	z
x	y'	z'
x	y	z'
x	y'	z

Figure 4.6. Generating a relation with a deletion violation

4.5. THE CASE OF FD AND MVD CONSTRAINTS

The relationship between update anomalies and normal forms is investigated in this section for the most general case where the set of dependencies includes both FDs and MVDs.

4.5.1. Insertion Anomaly and Normal Forms

In this section, we demonstrate that Theorems 4.16 and 4.17 can be extended to the FD and MVD case.

Theorem 4.19. *Let R be a relation scheme and Σ a set of MVDs and FDs which apply to R. If R is not in 4NF, then every nonempty relation $r \in SAT(\Sigma)$ has an IV.*

Proof. If R is not in 4NF then there exists a nontrivial MVD $X \rightarrow\rightarrow Y \in \Sigma^+$ where X is not a superkey. As per Theorem 4.2, let t be any tuple in r and let t^* be the tuple defined by $t^*[X] = t[X]$ and $t^*[A] \notin r[A]$ for all other attributes $A \in (R - X)$. The claim is that r has an insertion violation when t^* is inserted into it.

To verify this claim, the compatibility condition is satisfied because, from Lemma 4.1, $K - X \neq \emptyset$ for any candidate key K and so, from the construction of t^*, $t^*[K] \notin r[K]$. Also, $X \rightarrow\rightarrow Y$ is violated in $r \cup \{t^*\}$ for the reasons given in Theorem 4.16. □

Theorem 4.20. *A relation scheme R is in 4NF if and only if it has no IA.*

Proof. As for Theorem 4.17 and using Theorem 4.19. □

4.5.2. Deletion Anomaly and Normal Forms

In this section we show that Theorem 4.18 extends to the FD and MVD case provided that there is at least one pure MVD in the set of constraints. Before establishing the main results, we recall a preliminary result (Lemma 3.6) which we established in Chapter 3.

Lemma 4.21. *Let R be a relation scheme and let Σ be a set of FDs and MVDs which apply to R. If X is a set of attributes that is not a superkey, then there exists a relation containing at least two tuples that satisfies Σ and for which all tuples are identical on X.*

We now derive a lemma which will be used later to construct a relation with a DV.

Lemma 4.22. *Let R be a relation scheme and Σ a set of FDs and MVDs which apply to it. If $X \rightarrow\rightarrow Y$ is a pure MVD in Σ, then there exists a relation r which satisfies Σ and contains at least 4 distinct tuples ω_1, ω_2, ω_3 and ω_4 such that $\omega_1[X] = \omega_2[X] = \omega_3[X] = \omega_4[X]$, $\omega_1[Y]= \omega_2[Y]$, $\omega_2[Y]= \omega_3[Y]$, $\omega_3[Y] = \omega_4[Y]$, $\omega_1[Z] = \omega_3[Z]$, $\omega_3[Z] = \omega_2[Z]$, $\omega_2[Z] = \omega_4[Z]$ (where $Z = R - XY$).*

Proof. Form the tableau T_X as described in Section 2.5 and let $T^* = chase_\Sigma(T_X)$. The claim is that T^* satisfies the conditions of the theorem from which it follows trivially that so does $\rho(T^*)$ for any one-to-one valuation ρ. The desired property of T^* can perhaps be more easily illustrated by the figure below.

T*

	X	Y	Z
	.	.	.
ω_1:	x	y_1	z_1
ω_2:	x	y_2	z_2
ω_3:	x	y_1	z_2
ω_4:	x	y_2	z_1
	.	.	.

From Lemma 4.2, T^* consists of more than one row and every row is identical on X. From Lemma 2.2, one row in T^* is the row ω_d which contains only distinguished variables. For notational convenience in this proof, ω_d will be relabelled as ω_1. Next, by assumption $X \rightarrow Y \notin \Sigma^+$ since $X \rightarrow\rightarrow Y$ is pure and so, by Lemma 2.2, there must be at least one row in T^*, which we label as ω_2, which contains a nondistinguished variable in a Y-column and so $\omega_2[Y] = \omega_1[Y]$.

Suppose firstly that $\omega_1[Z] = \omega_2[Z]$. By Lemma 2.1, T^* satisfies $X \rightarrow\rightarrow Y$ and so there must exist a row ω_3 with $\omega_3[X] = \omega_1[X]$, $\omega_3[Y] = \omega_1[Y]$ and $\omega_3[Z] = \omega_2[Z]$ and a row ω_4 with $\omega_4[X] = \omega_1[X]$, $\omega_4[Y] = \omega_2[Y]$ and $\omega_4[Z] = \omega_1[Z]$. These conditions also imply that ω_1, ω_2, ω_3 and ω_4 are distinct and so satisfy the conditions of the theorem.

Alternatively, suppose that $\omega_2[Z] = \omega_1[Z]$. Since by Lemma 2.1 ω_1 contains only distinguished variables, $\omega_2[Z]$ contains only distinguished variables in the Z-columns. Hence since $X \rightarrow Z \notin \Sigma^+$ because $X \rightarrow\rightarrow Y$ is pure, there must exist a row ω_3 in Σ containing a nondistinguished variable in a Z-column and so $\omega_3[Z] = \omega_2[Z]$ and $\omega_3[Z] = \omega_1[Z]$ and hence ω_3 must be distinct from ω_2 and ω_1. There are then three exhaustive sub-cases to be considered: (i) $\omega_3[Y] = \omega_1[Y]$, (ii) $\omega_3[Y] = \omega_2[Y]$, (iii) $\omega_3[Y] = \omega_2[Y]$ and $\omega_3[Y] = \omega_1[Y]$. We consider each possibility in turn.

(i) $\omega_3[Y] = \omega_1[Y]$.

Since by Lemma 2.1T^* satisfies $X \to\to Y$, there must be a row ω_4 in T^* with $\omega_4[Z] = \omega_3[Z]$ and $\omega_4[Y] = \omega_2[Y]$. These conditions also imply that ω_4 is distinct and so ω_1, ω_4, ω_3 and ω_2 satisfy the requirements of the lemma.

(ii) $\omega_3[Y] = \omega_2[Y]$

Again, since T^* satisfies $X \to\to Y$, there must exist a distinct tuple ω_4 with $\omega_4[Y] = \omega_1[Y]$ and $\omega_4[Z] = \omega_3[Z]$. Again these conditions imply that ω_4 is distinct and so ω_1, ω_3, ω_4 and ω_2 satisfy the requirements of the lemma.

(iii) $\omega_3[Y] = \omega_2[Y]$ and $\omega_3[Y] = \omega_1[Y]$.

Again, to satisfy $X \to\to Y$, there must exist a row ω_4 with $\omega_4[Y] = \omega_1[Y]$ and $\omega_4[Z] = \omega_3[Z]$ and a row ω_5 with $\omega_5[Y] = \omega_2[Y]$ and $\omega_5[\mathrm{Z}] = \omega_3[\mathrm{Z}]$. Then ω_1, ω_5, ω_4 and ω_2 satisfy the requirements of the lemma. □

We now use this lemma to establish the main theorem of this section.

Theorem 4.23. *Let R be a relation scheme and let Σ be a set of FDs and MVDs which apply to R and such that Σ contains at least one pure MVD. R is in 4NF iff it has no DA.*

Proof.

Only if

As for Theorem 4.3.

If

We establish the contrapositive that if R is not in 4NF then it has a DA. Let $X \to\to Y$ be a pure MVD in Σ. X cannot be a superkey since this would imply that $X \to Y \in \Sigma^+$ thus contradicting the assumption that $X \to\to Y$ is pure. By Lemma

4.22, it follows that there is relation of at least four tuples with the specified properties, and this relation has a DV since deleting any of the specified tuples results in $X \to\to Y$ being violated. □

While it is easily seen that the if part of this theorem remains valid even when the restriction that Σ be pure is removed, the following example shows that the only if part does not remain valid if Σ is not pure. In other words, a relation scheme may not be in 4NF yet have no deletion anomaly.

Example 4.12. Let $R = \{A, B, C\}$ and let $\Sigma = \{A \to\to B, C \to B\}$. The only candidate key is AC and so both of the dependencies violate the 4NF condition. By inference rules A9 and A8 (see Chapter 2) it follows that Σ is equivalent to the set of FDs $\Sigma' = \{A \to B, C \to B\}$. But it follows directly from the definition of a deletion violation that if a relation has a deletion violation with respect to Σ then it must also have a deletion violation with respect to Σ'. However, as mentioned earlier, there can never be a deletion violation if only FDs are present and so R has no deletion anomaly. □

4.5.3. MA_1 and MA_2 Anomalies and Normal Forms

In this section we establish the important results that a relation scheme has an MA_1 anomaly if and only if it has an MA_2 anomaly and an absence of either anomaly is equivalent to 4NF. Before deriving these main results, we prove some preliminary lemmas.

The first lemma is an important result which will be used in the main theorems. Alternative proofs of the lemma have been given by Sagiv *et al.* [Sagiv et al. 1981] and by Vardi [Vardi 1988].

Lemma 4.24. *Let R be a relation scheme, let Σ be a set of FDs and MVDs which apply to R and let X be a set of attributes which is not a superkey. As before, denote the elements in DEP(X) by $\{X_1, \ldots, X_p, X_1^+, \ldots, X_j^+, W_1, \ldots, W_n\}$. Any relation of two tuples for which the two tuples are different on every attribute in one of the W's, say W_i, but equal on all other attributes satisfies Σ.*

Proof. For convenience, relabel the elements of DEP(X) so that W_i is relabelled as W_1. Consider now the satisfaction of the dependencies in Σ. Firstly, let $V \rightarrow Z$ be any FD in Σ and suppose to the contrary that it is violated in r. Let $W = X_1 \ldots X_p X_1^+ \ldots X_j^+ W_2 \ldots W_n$. By the construction of r, for $V \rightarrow Z$ to be violated one must have that $V \subseteq W$ and $Z \cap W_1 \neq \emptyset$. From $V \rightarrow Z$ and inference rules A5 and A8, it follows that $W \rightarrow\rightarrow Z$ and combining with the fact that $X \rightarrow\rightarrow W$ from rule A11, and using the transitivity rule A6, gives $X \rightarrow\rightarrow Z - W$ which is the same as $X \rightarrow\rightarrow Z \cap W_1$ since DEP(X) covers R. Applying rule A10 to $V \rightarrow Z$ shows that $V \rightarrow Z \cap W_1$. Then, since $V \subseteq W$ and the fact that W and W_1 are disjoint, it follows that V and $Z \cap W_1$ are disjoint and so, by rule A9, $X \rightarrow Z \cap W_1$ and thus, by property (v) of DEP(X) (refer to Section 2.4.2), $Z \cap W_1 \subseteq X_1 \ldots X_p X_1^+ \ldots X_j^+$. This is a contradiction since $Z \cap W_1$ is a nonempty subset of W_1 but, by property (ii) of DEP(X), W_1 is disjoint from $X_1 \ldots X_p X_1^+ \ldots X_j^+$.

The next part of the proof is based on the method used by Beeri *et al.* [Beeri et al. 1977]. Consider now the case of any MVD $V \rightarrow\rightarrow Z$ in Σ and suppose to the contrary that it is violated. From the definitions of t_1 and t_2, the only way for this to happen is for $V \subseteq W$, $Z \cap W_1 \neq \emptyset$ and $(R - VZ) \cap W_1 \neq \emptyset$. We can note that these conditions imply that $Z \cap W_1$ is a proper subset of W_1. Since $V \subseteq W$, it follows by rule A5 that $W \rightarrow\rightarrow Z \in \Sigma^+$. Also, by rule A11, $X \rightarrow\rightarrow W \in \Sigma^+$ and so using the transitive rule $X \rightarrow\rightarrow Z - W$, that is $X \rightarrow\rightarrow Z \cap W_1 \in \Sigma^+$. This is a contradiction since by property (iii) of DEP(X), $Z \cap W_1$ must be a union of elements in DEP(X) but this cannot occur since by property (ii) the sets in DEP(X) are disjoint and $Z \cap W_1$ is a nonempty proper subset of W_1. Hence r satisfies $V \to\to Z$ and the result is established. □

We now illustrate this result by an example taken from the book by Paredaens *et al.* [Paredaens et al. 1989].

Example 4.13. Let $R = \{A, B, C, D, E, F, G\}$ and $\Sigma = \{AB \to\to DE, E \to\to F, E \to C\}$. Standard algorithms [Beeri 1980] can be used to show that if $X = AB$, then $X^+ = \{A, B, C\}$ and DEP(X) = $\{A, B, C, DE, F, G\}$. The relation shown in Figure 4.7 satisfies Σ. □

A	B	C	D	E	F	G
1	1	1	1	1	1	1
1	1	1	0	0	1	1

Figure 4.7. An example illustrating Lemma 4.24

We now use this result to derive another important lemma.

Lemma 4.25. *Let R be a relation scheme, let Σ be a set of FDs and MVDs which apply to R and let X be a set of attributes which is not a superkey. If K is a candidate key for R and W is a set in DEP(X) such that* $W \cap X^+ = \emptyset$ *then* $K \cap W \neq \emptyset$.

Proof. By Lemma 4.24, any two tuple relation for which the tuples are the same except for those attributes in W is in SAT(Σ). This implies that the relation is also in SAT(Σ_K) and hence the two tuples must be distinct on every candidate key from which the result follows immediately. □

This result also has another interesting corollary concerning the relationship between the dependency basis and the structure of candidate keys. Let us denote the number of sets in DEP(X) which are disjoint from X^+ by $|\mathrm{DEP}(X)|$.

Lemma 4.26. *If R is a relation scheme, Σ a set of FDs and MVDs which apply to R and X a set of attributes which is not a superkey, then every candidate key K contains at least* $|DEP(X)|$ *attributes.*

Proof. Follows directly from Lemma 4.25 and the fact that the *W's* in DEP(X) are disjoint (refer to Section 2.4.2). □

We now present one of the main results of this section which shows that 4NF is equivalent to the condition that a relation scheme have no MA_2.

Theorem 4.27. *If R is a relation scheme and Σ is a set of FDs and MVDs that apply to R and contains at least one nontrivial FD, then R is in 4NF iff it has no* MA_2*.*

Proof.

Only if

As for Theorem 4.3.

If

We shall show the contrapositive that if R is not in 4NF then it has an MA_2. We firstly note that since candidate keys and dependency violation are the same for equivalent sets of dependencies, then the definition of an MA_2 is independent of which equivalent set of dependencies is used and since, as shown in Chapter 2, every set of FDs and MVDs has a pure cover then, without loss of generality, Σ can be assumed to be pure. If Σ contains only FDs, then Theorem 4.12 shows that R has an MA_2. Alternatively, if there is an MVD $X \rightarrow\rightarrow Y$ in Σ, then X cannot be a superkey since this would imply that $X \rightarrow Y \in \Sigma^+$ which violates the assumption that Σ is pure. We shall now show that there exists a relation which has an MV_2. In the following proof, we often want to consider the complement of an MVD as well and so $X \rightarrow\rightarrow Y$ will often be written as $X \rightarrow\rightarrow Y \mid Z$ where $Z = R - XY$.

Let K be the primary key of R. Split each of the sets X, Y, and Z into a set which intersects with K and a set which is disjoint from K and thus write the MVD $X \rightarrow\rightarrow Y \mid Z$ as $X'X_k \rightarrow\rightarrow Y'Y_k \mid Z'Z_k$ where $X = X'X_k$, $Y = Y'Y_k$, $Z = Z'Z_k$ and $X'Y'Z' \cap K = \emptyset$. Also, since by assumption there is at least one nontrivial FD $X \rightarrow A \in \Sigma$, it follows that $R - A$ is a superkey and so R cannot be a candidate key and hence every candidate key must be a proper subset of R. Thus at least one of the sets X', Y' and Z' must be nonempty. We now consider several cases separately.

(a) $X_k = \emptyset$.

Define a relation r of two tuples, t_1 and t_2, such that the two tuples are different on every attribute. Obviously, $r \in$ SAT(Σ). Then define the tuple t^* by $t^*[X] = t_1[X]$, $t^*[R\text{-}X] = t_2[R - X]$. The claim is that r has an MV_2 when t_2 is replaced by t^*. Condition (i) of an MV_2 follows from the definition of r. Since X is not a superkey then, by Lemma 4.1, $K' - X \neq \emptyset$ for any candidate key K' and so from the definitions of t^* and r it follows that $t^*[K'] \neq t_1[K']$ and thus (ii) holds. Also, $t^*[K] = t_2[K]$ since $X_k = \emptyset$ and so (iii') is satisfied. Condition (iv) follows from the fact that $Z \neq \emptyset$ (since $X \rightarrow\rightarrow Y$ is nontrivial) and from the definitions of r and t^*.

(b) $X_k \neq \emptyset$.

Firstly, we note that in this case neither Y_k nor Z_k can be empty. To verify this, assume firstly that Y_k is empty. Since $K \rightarrow Y \in \Sigma^+$ because K is a candidate key, combining this with $X \rightarrow\rightarrow Y$ and using rule A9 implies that $X \rightarrow Y \in \Sigma^+$ thus contradicting the assumption that Σ is pure. Similarly, if Z_k is empty then it follows that

$X \rightarrow Z \in \Sigma^+$ which again contradicts the assumption that Σ is pure. We now break the proof up into several sub cases.

(b.1) Either $Y' \neq \emptyset$ or $Z' \neq \emptyset$.

Assume that $Y' \neq \emptyset$. By symmetry, exactly the same argument will apply if $Z' \neq \emptyset$. Since $X \rightarrow\rightarrow Y$ is nontrivial and Σ is pure, we firstly claim that there exists a $W \in$ DEP(X) which is disjoint from Y and from X^+. To establish this, we suppose that it is not the case and derive a contradiction. Let $W_1, \ldots, W_n$ be the sets in DEP(X) which are disjoint from X^+. Since $X \rightarrow\rightarrow Y \in \Sigma$, it follows from the properties of DEP(X) that Y is a union of elements from DEP(X) and, since by assumption there is no W_i disjoint from Y, then $W_1 \ldots W_n \subseteq$ Y. It follows that $Z = R - XY \subseteq X_1^+ \ldots X_j^+$ and thus $X \rightarrow Z$ is a nontrivial FD in Σ^+. This contradicts the assumption that Σ is pure and so there exists a W_i in DEP(X) which is disjoint from X^+ and Y.

Construct a relation r of two tuples, t_1 and t_2, which are identical on all attributes except those in W and modify r by replacing t_2 by the tuple t^* where $t^*[Y'] \neq t_1[Y']$ and t^* and t_2 are identical on all other attributes. The claim is that r has an MV_2 when t_2 is replaced by t^*. Condition (i) of an MV_2 follows from Lemma 4.24. Next, let K' be any candidate key in R. If $K' \cap Y' = \emptyset$, then $t^*[K'] = t_2[K']$ from the definition of t^* and thus $t^*[K'] \neq t_1[K']$ since $r \in$ SAT(Σ_k). Alternatively, if $K' \cap Y \neq \emptyset$, then $t^*[K'] \neq t_1[K']$ from the definition of t^*. Thus the compatibility condition (ii) is satisfied. Condition (iii') holds because t^* and t_2 differ only on the attributes in Y', and (iv) is valid since the tuples t^* and t_1 agree on X but differ on Y and Z.

(b.2) $Y' = \emptyset$ and $Z' = \emptyset$.

In this case we can write the MVD $X \rightarrow\rightarrow Y$ as $X'X_k \rightarrow\rightarrow Y_k \mid Z_k$. We shall consider separately the two possibilities: (1) there exists a dependency in Σ with a subset of X_k as the left-hand side; (2) there does not exist such a dependency.

(b.2.1) There exists a dependency in Σ with a subset of X_k as the left-hand side.

In other words, there exists either $X_1 \rightarrow\rightarrow V$ or $X_1 \rightarrow A \in \Sigma$ with $X_1 \subseteq X_k$.

Consider firstly the MVD case. As mentioned previously, since there is at least one FD in Σ, no candidate key covers R and since $X_1 \subset K$, either $V - K' \neq \emptyset$ or $R - X_1V - K \neq \emptyset$. Then the same construction used in case (b.1) shows that R has an MA_2.

Consider next the FD case. Firstly, by Lemma 4.9, A cannot be a member of K because X_1 is assumed to be a subset of K. Then, since X is not a superkey, X_1 is not a superkey and so there exists a nonempty W in DEP(X_1) such that $W \cap X_1^+ \neq \emptyset$. Choose arbitrarily any such W and construct a relation r of two tuples, t_1 and t_2, for which the two tuples are identical on every attribute except for those in W. Modify r by replacing t_2 with the tuple t^* which is defined by $t^*[A] \neq t_1[A]$ and t^* and t_2 are identical for all other attributes. The claim is that r has an MV_2 when t_2 is replaced by t^*. Condition (i) follows from Lemma 4.24. The compatibility condition (ii) holds since, by Lemma 4.25, t_1 and t_2 differ on $K' \cap W$ for any candidate key K', and so t^* and t_1 differ on K' since $t^*[W] = t_2[W]$. Condition (iii') follows because t_2 and t^* differ only on A and, as indicated earlier, A is not part of the primary key, while (iv) holds because t^* and t_1 agree on X_1 yet differ on A.

(b.2.2) There doesn't exist a dependency in Σ with a subset of X_k on the left-hand side.

Firstly, since R is not a candidate key, it follows that X' in the MVD $X'X_k \rightarrow\rightarrow Y_k \mid Z_k$ is nonempty.

Construct a relation r of two tuples, t_1 and t_2, for which the two tuples are identical on every attribute in X_k and different on all other attributes. Define the tuple t^* by $t^*[X] = t_1[X]$ and for all other attributes t^* is identical to t_2. We claim that r has an MV_2 when t_2 is replaced by t^*. Firstly, $r \in$ SAT(Σ) because t_1 and t_2 are distinct on all attributes except those in X_k and so the only dependency that r could violate is one with a subset of X_k on the left-hand side and by (b.2.2) this cannot occur. So condition (i) of an MV_2 is satisfied. Next, since $X'X_k$ is not a superkey, it follows from Lemma 4.1 that, for every candidate key K', $K' - X'X_k \neq \emptyset$ and so by definition of r and t^*, $t^*[K'] \neq t_1[K']$ and thus the compatibility condition (ii) holds. Condition (iii') is satisfied because t^* and t_2 differ only on X' and (iv) is satisfied because t^* and t_1 agree on X yet differ on Y and Z.

□

The requirement in the theorem that the set of dependencies must contain at least one FD is necessary for the equivalence of 4NF and MA_2. If we drop this requirement, in other words there are only MVDs in the set of dependencies, then the only candidate key is R since only trivial FDs are implied by a set of MVDs [Maier 1983]. So every attribute in R is prime and it follows from the definitions of modification violations that R has no MA_2. However, any nontrivial MVD violates the 4NF condition since R is the only candidate key and so a relation scheme with only MVDs in the set of dependencies is not in 4NF yet has no MA_2. A simple corollary of the theorem is the following important result that 4NF is also equivalent to an absence of an MA_1 anomaly.

Corollary 4.28. *If R is a relation scheme and Σ is a set of FDs and MVDs that apply to R and contains at least one nontrivial FD, then R is in 4NF iff it has no MA_1.*

Proof.

If

The contrapositive - that if R is not in 4NF then it has an MA_1 anomaly - follows since by the theorem, if R is not in 4NF then it has an MA_2 anomaly, and it follows directly from the definitions of the anomalies that if a relation scheme has an MA_2 anomaly then it must have an MA_1 anomaly.

Only If

As for Theorem 4.3. □

4.5.4. MA_3 Anomaly and Normal Forms

We now address the problem of deriving necessary and sufficient conditions for a relation scheme to have no MA_3. We proved earlier (Theorem 4.14) that in the case where the set of constraints contains only FDs, the necessary and sufficient condition for no MA_3 is a weaker condition than BCNF. The following example shows that similarly, in the case where the set of constraints includes both MVDs and FDs, 4NF is a stronger condition than is required for a relation scheme to have no MA_3.

Example 4.14. Consider the following relation scheme $R = \{G, S, T, H\}$. The meaning of the attributes are: G - represents the name of a tutorial group, S - represents the name of a student, T - represents the name of a tutor, H - represents the time in the week that a tutorial group meets. A tuple $<g, s, t, h>$ represents the information that a student s attends a tutorial group g which is conducted by a tutor t and the tutorial takes place at hour h. Suppose also that the following rules apply. Each tutorial group can contain several students and students may belong to several tutorial groups. Each tutorial group has only one tutor and a tutor may be a tutor to only one group. A tutorial group may meet several times week with the same set of students and the same tutor. An example relation is illustrated in Figure 4.8.

From these rules the following set of dependencies can be derived $\{G \rightarrow T, T \rightarrow G, G \rightarrow\rightarrow H\}$. It can be verified that the candidate keys are GSH and HST and that the MVD $G \rightarrow\rightarrow H$ is pure. R is not in 4NF because none of the left-hand sides is a superkey yet it has no MA_3 anomaly because every attribute in R is prime. □

G	S	T	H
Phyics1	Jones	Fermi	Mon - 10.00
Physics1	Smith	Fermi	Tue - 2.00
Database1	Jones	Codd	Tue - 2.00
Physics1	Jones	Fermi	Tue - 2.00
Physics1	Smith	Fermi	Mon - 10.00

Figure 4.8. A relation with no MV_3

We now present the main theorem of this section which gives a necessary and sufficient condition for a relation scheme to have no MA_3.

Theorem 4.29. *Let R be a relation scheme and let Σ be a set of FDs and MVDs that apply to R and which contains at least one pure MVD. R has no MA_3 iff every attribute in R is prime.*

Proof.

If

Let $r(R)$ be any relation in SAT(Σ). Since every attribute in R is prime, this implies that any modification which leaves the prime attributes of a tuple unchanged doesn't change the tuple and so r has no MV_3 violation since $r \in$ SAT(Σ).

Only If

We shall establish the result by showing the contrapositive that if there is a nonprime attribute in R then R must have an MA_3 anomaly. Firstly, it follows directly from the definition of the modification violations and modification anomalies that if a relation scheme has a modification anomaly (of any type) with respect to one set of dependencies, then it must also have a modification anomaly (of the same type) with respect to any equivalent set of dependencies. Since, as outlined in Section 2.4.3 , every set of FDs and

MVDs has a pure reduced cover then, without loss of generality, Σ is assumed to be pure and reduced. Also, if the original set of dependencies Σ contains a pure MVD then a pure reduced cover for Σ must contain at least one MVD. To establish this, let $X \rightarrow\rightarrow Y$ be a pure MVD in Σ and suppose to the contrary that Σ_1, a pure reduced cover for Σ, contains only FDs. By a well known result (Theorem 7.2 in Maier [Maier 1983]), there must exist an FD $X \rightarrow Y$ or an FD $X \rightarrow R - XY$ in Σ_1, which are also in Σ^+ since Σ_1 is a cover for Σ, thus contradicting the assumption that $X \rightarrow\rightarrow Y$ is a pure MVD.

We shall now prove that R has an MA_3 by showing that there always exists a relation defined over R that has an MV_3. We note firstly that, because Σ is pure, for any MVD $X \rightarrow\rightarrow Y$ in Σ, X cannot be a superkey since if it was it would imply that $X \rightarrow Y \in \Sigma^+$ thus contradicting the assumption that Σ is pure. Let $X \rightarrow\rightarrow Y$ be an MVD in Σ. Since $XYZ = R$ and by assumption R contains a nonprime attribute, there are only two cases to consider: (a) YZ contains a nonprime attribute or (b) only X contains a nonprime attribute. We now consider each case in turn.

(a) YZ contains a nonprime attribute.

In this case, the same construction as used in case (b.1) of Theorem 4.27 demonstrates that R has an MA_3.

(b) Only X contains a nonprime attribute.

In this case we write X as $X'X_p$ where X' contains the nonprime attributes in X and X_p contains the prime attributes. We firstly show that we can restrict our attention to the case where the dependencies in Σ have the following property that X' is a subset of the left-hand side of every MVD in Σ. If this property is not satisfied then, since X' contains all the nonprime attributes in R, this implies that either the right-hand side of an MVD, or its complement, contains nonprime attributes and so the same argument as in case (a) above shows that R has an MA_3.

We now want to show that there exists a set W in $\mathrm{DEP}(X_p)$ such that W is disjoint from X_p^+ and W has nonempty intersection with each of the sets X', Y and Z. The proof of this assertion is rather lengthy and accordingly we present it as a separate following lemma (Lemma 4.30). Construct then a relation r of two tuples, t_1 and t_2, for which the two tuples are different on every attribute in W and equal on all other attributes. Replace t_2 by the tuple t^* which is defined by $t^*[X'] = t_1[X']$ and $t^*[R - X'] = t_2[R - X']$. The claim is that r has an MV_3 when t_2 is replaced by t^*. Condition (i) follows from Lemma 4.24. Condition (ii) follows from the fact that r satisfies the key constraints and t^* and t_2 are equal on prime attributes, which also implies condition (iii"). Finally, condition (iv) follows from the fact that W has nonempty intersection with both Y and Z and so by definition of r and t^*, t_1 and t^* agree on X but differ on both Y and Z. □

In order to derive the lemma needed for the completion of the theorem, we need the following algorithm, taken from and based on that given by Beeri [Beeri 1980] and later by Paredaens *et al.* [Paredaens et al. 1989], which correctly calculates DEP(X). Our version is simpler than that provided by Paredaens *et al.* since we assume the set of constraints to be reduced and thus each FD contains only a single attribute on the right-hand side.

ALGORITHM 4.2.

INPUT: A relation scheme R, a reduced set of FDs and MVDs, Σ, which apply to R and a set of attributes X.

OUTPUT: DEP(X)

METHOD:

var : U', U'', V', V'', W : sets of attributes;
OLD_DEP, NEW_DEP : sets of sets of attributes;

1: NEW_DEP := $\{A \mid A \in \mathrm{X}\} \cup \{R - X\}$;
repeat
2: **for each** $U \rightarrow\rightarrow V$ **or** $U \rightarrow V \in \Sigma$ **do**
3: $U'' := \cup\{W \mid (W \in$ NEW_DEP) and $(W \cap U \neq \emptyset)\}$;
4: $V'' := V - U''$;
5: **if** $V'' \neq \emptyset$
then
for each $W \in$ NEW_DEP **do**
6: **if** $(W \cap V'' \neq \emptyset)$ **and** $(W \cap V'' \neq W)$
then
7: NEW_DEP := (NEW_DEP - $\{W\}) \cup$
$\{W \cap V'', W - V''\}$;
od
od
until (NEW_DEP = OLD_DEP);

We now use the algorithm in the following lemma.

Lemma 4.30. *Let X' be the set of all nonprime attributes in a relation scheme R and let Σ be a pure reduced set of FDs and MVDs such that the left-hand side of every MVD Σ contains X'. Then, for any MVD $X'X \rightarrow\rightarrow Y|Z$ in Σ, DEP(X) = $\{X_1, \ldots, X_n, X^+_1, \ldots, X^+_j, W\}$ where $X = X_1 \ldots X_n$ and $X^+_1, \ldots, X^+_j$ are single attribute sets in X^+ and $W = X'Y'Z'$ where Y' and Z' are nonempty subsets of Y and Z respectively.*

Proof. We shall prove the result by induction on the steps in Algorithm 4.2 used for generating DEP(X) by showing that if OLD_DEP in the algorithm has the stated property in the lemma at the start of the repeat loop, then the new basis, NEW_DEP, will have the same property at the end of the loop. Initially, because $R = X'XYZ$, OLD_DEP is set to $\{X_1, \ldots, X_n, X'YZ\}$ where $X = X_1 \ldots X_n$ and so obviously the property is satisfied initially. Then assume inductively that OLD_DEP can be written as $\{X_1, \ldots, X_p, X_1^+, \ldots, X_j^+, X'Y'Z'\}$. We consider separately the two cases of whether the dependency that is tested at line 2 of Algorithm 4.2 is an MVD or an FD.

(a) The MVD case

By the assumption of the lemma, any MVD $U \rightarrow\rightarrow V \in \Sigma$ can be written as $X'U' \rightarrow\rightarrow V$ where $U = X'U'$. Consider $V'' = V - U''$ which is defined at line 4 of Algorithm 4.2. Obviously the set $X'Y'Z'$ is a subset of U'' since $X' \cap U = X'$ and so V'' must be disjoint from $X'Y'Z'$ and thus, by the inductive assumption, it must be a subset of the set $X_1 \ldots X_p X_1^+ \ldots X_j^+$. Thus the test at line 6 of the algorithm fails since if $W = X'Y'Z'$ then the first condition fails, and if W is any other set in NEW_DEP then, by the induction hypothesis, it contains only one attribute and so if the first condition in line 6 succeeds then the second one must fail. Hence no change is made to NEW_DEP and so the inductive property still holds.

(b) The FD case

Consider the effect on NEW_DEP if instead the FD $U \rightarrow V$ is applied at line 2. Since Σ is reduced, V consists of a single attribute. We shall break the proof up into the following exhaustive sub cases.

(b.1) $V \notin X'Y'Z'$

If $V \notin X'Y'Z'$ then, by the inductive assumption, V must be a single attribute set in NEW_DEP and thus the test at line 6 will again fail for the same reasons as in case (a) and so NEW_DEP is unaltered.

(b.2) $V \in X'$

We shall show that NEW_DEP is unchanged by assuming the contrary and deriving a contradiction. Since V is a single attribute and NEW_DEP is updated only if the tests at lines 5 and 6 succeed, then $V'' \neq \emptyset$ and thus $V'' = V$ since V is a single attribute. Thus if $W \cap V'' \neq \emptyset$ then $W \cap V'' = V$ and so if the test at line 6 succeeds then line 7 is executed and a set consisting of V alone is added to NEW_DEP. If this happens then, since at any stage of Algorithm 4.2 and for any set Z in NEW_DEP, $X \rightarrow\rightarrow Z \in \Sigma^+$, it follows that $X \rightarrow\rightarrow V \in \Sigma^+$ and so, since the left and right-hand sides of MVDs in Σ are disjoint, a simple application of the inference rules shows that the MVD $(X' - V)\ X \rightarrow\rightarrow Y \in \Sigma^+$ which contradicts the assumption that Σ is reduced. Thus NEW_DEP remains unchanged.

(b.3) $V \in Y'$.

From the same argument as in case (b.2), if NEW_DEP is modified then V becomes an element of NEW_DEP and $X \rightarrow\rightarrow V \in \Sigma^+$. We now want to verify that the inductive hypothesis remains valid. If $V \in X^+$ then the hypothesis is still true so assume to the contrary that $V \notin X^+$. It can be easily seen from Algorithm 4.2 that once V becomes a member of NEW_DEP, it will remain a member and so V is a member of DEP(X) and is disjoint from X^+. From Lemma 4.24, any two tuple relation r which is identical on every attribute in R except V is in SAT(Σ). Applying Lemma 4.25 then shows that V must be a member of every candidate key. However, since we are assuming there is an FD $U \rightarrow V$ in Σ, then $R - V$ is superkey and so contains at least one candidate key which is disjoint from V and thus contradicting the fact V is a member of every candidate key. So $V \in X^+$.

It remains to verify the other inductive property that the sets X', Y' and Z' remain nonempty if V is added to NEW_DEP. If V becomes a member of NEW_DEP, then by line 7 the set $X'Y'Z'$ in NEW_DEP is replaced by $X'Z'Y' - V$. Since by the induction hypothesis $Y' \subseteq Y$, then $Y' - V \subseteq Y$ and so it suffices to show that $Y' - V \neq \emptyset$.

Suppose to the contrary that $Y' - V = \emptyset$. If this happens, then from the argument in the previous paragraph and the induction hypothesis, every attribute in Y must be in X^+ and so $X \to Y$ is in Σ^+ and a simple application of the inference rules shows that $X'X \to Y$ is in Σ^+. However this contradicts the assumption that Σ is pure and so $Y' - V \neq \emptyset$.

(b.4) $V \in Z'$

Same argument as in case (b.3) □

We note that in the case that Σ does not contain a pure MVD (so Σ is equivalent to a set of FDs), then the necessary and sufficient condition reduces to the one given in Theorem 4.14.

4.6. RELATED WORK AND CONCLUSIONS

The results in this chapter build upon the work of Fagin [Fagin 1979; Fagin 1981] and in this section we review his results and summarise the extension to those results contained in this chapter. In his earlier work [Fagin 1979], Fagin showed that both BCNF, 4NF and PJNF all have a similar structure in that each of the normal forms is equivalent to the condition that any relation satisfying the key constraints must also satisfy a set of constraints of the corresponding type (FDs for BCNF, FDs and MVDs for 4NF, FDs, MVDs and JDs for PJNF). In fact, for the case of PJNF, this property is the definition proposed by Fagin, whereas in the case of BCNF and 4NF it was proven to be equivalent to the usual definitions. An immediate consequence of these results is that a relation scheme is in the appropriate normal form if and only if it has no insertion anomaly with respect to a set of dependencies of the corresponding type. Thus the traditional normal forms can be justified on the grounds that they are precisely the conditions which ensure that, by only checking key uniqueness, the satisfaction of constraints can be guaranteed after an insertion into a relation.

In his later paper [Fagin 1981] and as discussed in Chapter 2, Fagin generalised the types of allowable constraints to include domain constraints and any sentence in first-order logic involving the attribute names. Motivated by his earlier results just discussed, he defined the normal form DK/NF (refer to Chapter 2) and proved that a relation scheme is in DK/NF if and only if it has no insertion anomaly and no deletion anomaly. In the case where the only constraints are MVDs, FDs and key constraints, this result reduces to the statement that a relation scheme is in 4NF if and only if it has no insertion anomaly. In this chapter we have extended this result for the case where the types of constraints permitted are only FDs and MVDs. For the case of an insertion anomaly, whereas Fagin's results showed that every relation scheme which is not in BCNF or 4NF must have at least one insertion violation, we have shown the stronger result (Theorems 4.2, 4.16, 4.19) that every legal relation defined over the scheme which is not in BCNF or 4NF has an insertion violation.

The second contribution of this chapter has been to provide a characterisation of 4NF in terms of the absence of a deletion anomaly. It was proved (Theorem 4.18) that in the case of MVD constraints alone, 4NF is equivalent to no deletion anomaly; whereas in the case of FD and MVD constraints we showed that the same result is still valid (Theorem 4.23) provided that there is at least one pure MVD (see Chapter 2) in the set of constraints.

The other main contribution of this chapter has been to define a new type of key-based update anomaly, called a modification anomaly, and derive necessary and sufficient conditions for its absence in a relation scheme. We defined a modification anomaly as occurring if the modification of a tuple in a relation results in the constraints being violated while key uniqueness is satisfied and the identity of the tuple is preserved. According to three different interpretations of what is meant by preserving the identity of a tuple, three different types of modification anomalies were defined. For the case of FD constraints, the main results that we have established are that for two of the modification anomaly types, the anomalies are equivalent conditions on a relation scheme and they are absent if and only if the scheme is in BCNF (Theorems 4.7 and 4.12). However, for the third type of modification anomaly, we proved (Theorem 4.14) that it is absent in a relation scheme if and only if it satisfies a new condition that we call prime attribute normal form (PANF). The PANF condition requires that the relation scheme be in 3NF and all the attributes in the left-hand side of every FD be prime. PANF is thus a stronger condition than 3NF but weaker than BCNF. We then showed (Theorem 4.15) that PANF is not comparable to EKNF, another improvement of 3NF, and that a synthesis algorithm can be used to generate relation schemes which are in PANF. The relationship between normal forms and the key-based update anomalies for the FD case is illustrated in the figure below.

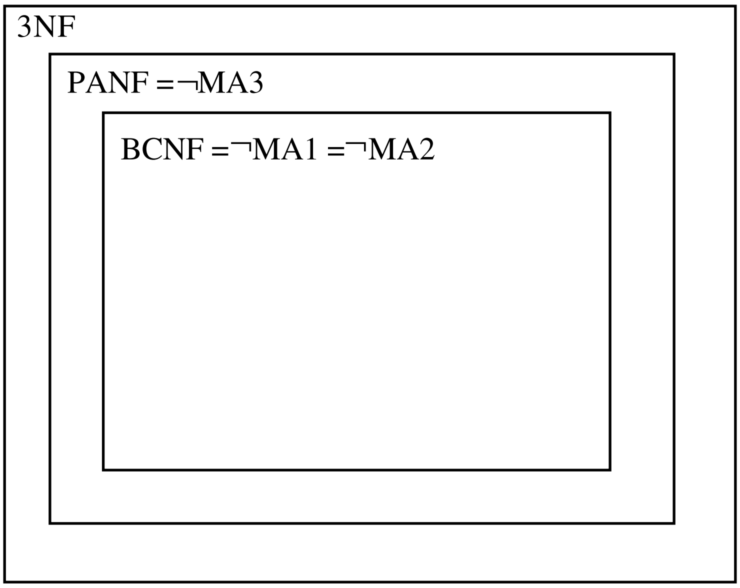

Figure 4.8. Relationship between normal forms in the FD case

We then showed that for the more general case where the set of constraints contains both FDs and MVDs, similar results to those obtained for the FD case apply. In particular, 4NF is equivalent to no MA_1 and MA_2 in a relation scheme provided that the set of constraints contains at least one FD (Theorems 4.26 and 4.27), whereas for MA_3, the equivalent condition is that every attribute in the relation scheme be prime provided that the set of dependencies contains at least one pure MVD (Theorem 4.28). The relationship between 4NF and these update anomalies is shown below for the FD and MVD case.

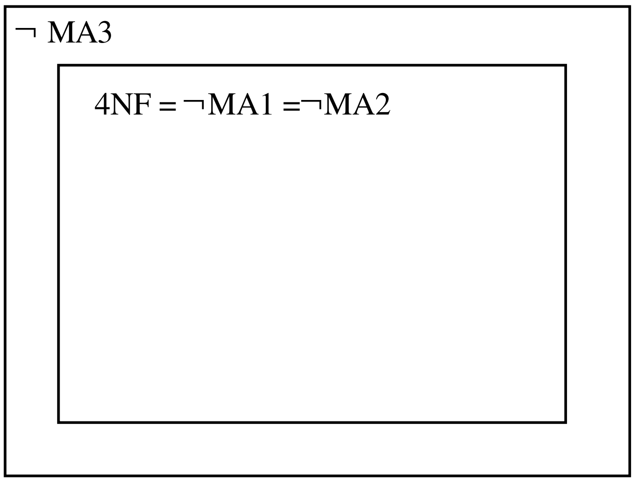

Figure 4.9. Relationship between normal forms in the FD and MVD case

CHAPTER 5

UNPREDICTABLE INSERTIONS AND NORMAL FORMS

5.1. INTRODUCTION

In this chapter normalisation shall be viewed from another perspective, that of ensuring that updates to relations do not behave in an *unpredictable* manner. This approach is originally due to Bernstein and Goodman [Bernstein and Goodman 1980] and later reformulated in a slightly different fashion by Vossen [Vossen 1988]. Both of these works considered only the case of FD constraints and it was shown that for this case, BCNF is a necessary and sufficient condition for predictable behaviour in updates to relations. In this chapter we extended their approach to include MVD constraints. The results of this chapter will also appear in the literature [Vincent and Srinivasan 1994d].

Firstly, let us explain what is meant by an *unpredictable update*[7] by the following example taken from Codd's original paper [Codd 1972] on normalisation. Let the set of attributes be $\{E\#, JC, D\#, M\#, CT\}$ and the set of FD constraints be $\{E\# \rightarrow JC\ D\#\ M\#\ CT, D\# \rightarrow M\#\ CT, M\# \rightarrow D\#\ CT\}$ where the meanings of these attributes are as follows: *E#* - employee number, *JC* - job code, *D#* - department number, *M#* - employee number of manager and *CT* - contract type. The relation shown in Figure 5.1 satisfies the set of constraints.

---

[7]The terminology used here differs from the terminology used by the authors just mentioned. What is called an *upredictable insertion* in this chapter is referred to as an *insertion anomaly* by those authors. The reason for adopting a different terminology is to avoid confusion with the update anomalies defined in other chapters.

E#	JC	D#	M#	CT
1	a	x	11	g
2	c	x	11	g
3	a	y	12	n
4	b	x	11	g
5	b	y	12	n
6	c	y	12	n
7	a	z	13	n
8	c	z	13	n

Figure 5.1. An example illustrating an unpredictable insertion

If one considers insertions for the moment then, intuitively speaking, a relation scheme has an *unpredictable insertion* if an insertion into one relation defined over the scheme requires information to be supplied for one set of attributes, while an insertion into a different relation requires information to be supplied for a different set of attributes. For example, if in Figure 5.1 one wants to insert the fact that an employee with $E\# = 9$ and $JC = d$ is to work in $D\# = x$ then, to satisfy the FD constraints, the values of *M#* and *CT* are already determined by the tuples in the relation and so the inserted tuple must be $<9, d, x, 11, g>$. In this case, no new information has been supplied for *M#* and *CT*. If instead one wants to insert the information that an employee with $E\# =10$ and $JC = c$ is to work in a new department with $D\# = w$, then new information has to be supplied concerning the *M#* and *CT#* of department *w* since there is no tuple with a department *w* in the relation. So the relation scheme is considered to have an unpredictable insertion since the two different insertions require different information to be supplied, in other words the effect of an insertion is not predictable. To be more precise and using the terminology of Bernstein and Goodman, the insertion of a tuple *t* into a relation *r* *affects* a set of attributes *X* if the projection of *r* onto *X* is not equal to the projection of $r \cup \{t\}$

onto X, and conversely is *unaffected* if the projections are the same. A relation scheme is then defined to have an unpredictable insertion if there exists at least two different insertions on relations defined over the scheme such that the attributes in an FD constraint are affected by one of the insertions but not by the other. If one notes that every FD is affected by the insertion of a tuple into a relation which is distinct on every attribute, then the unpredictable insertion condition can be simplified to the requirement that there exists an insertion which does not affect an FD.

We adopt a similar approach for the case of MVD constraints. Given a relation scheme R and a set Σ of FD and MVD constraints, R is defined to have an *unpredictable insertion* if there exists a nontrivial dependency, either $X \rightarrow Y$ or $X \rightarrow\rightarrow Y$, in Σ and a relation r defined over R and a tuple t such that r and $r \cup \{t\}$ have the same projection onto XY. A relation scheme is then defined to be in *insertion normal form* if it doesn't have an unpredictable insertion. However, as described in Chapter 3, there are three equally plausible choices for the set of constraints - the set Σ of FDs and MVDs derived from the database design; Σ plus all MVDs $X \rightarrow\rightarrow R - XY$ corresponding to the MVDs $X \rightarrow\rightarrow Y$ in Σ; and lastly, Σ^+, the set of all nontrivial FDs and MVDs implied by Σ. We allow for each of these possibilities and define an insertion normal form corresponding to each case as INF_1, INF_2 and INF_3.

The structure of this chapter and the main results obtained in it are as follows. In Section 5.3, the relationship between the insertion normal forms and 4NF is investigated for the case where the only constraints are MVDs. The main results derived are that INF_1, INF_2 and INF_3 are all equivalent to each other and to 4NF. In Section 5.4, the investigation is extended to the case of FD and MVD constraints. Somewhat surprisingly, for this case, although INF_2 and INF_3 are again equivalent to each other and to 4NF, INF_1 is a weaker condition; in other words, 4NF is a stronger condition than is necessary to avoid the presence of update anomalies if the only facts allowed are those which correspond to the dependencies in Σ. Finally, Section 5.5 contains concluding remarks. The relationship between the insertion normal forms and BCNF for the case of

FD constraints alone is not investigated in this chapter since, as discussed previously, it has already been shown that the insertion normal forms and BCNF are equivalent in this case [Bernstein and Goodman 1980; Vossen 1988].

5.2. THE DEFINITIONS OF UNPREDICTABLE INSERTIONS

In this section we present formal definitions of unpredictable insertions and the corresponding normal forms in which these difficulties are absent. Firstly, the only insertions of interest are those which result in a relation being changed from one valid state to another valid, but different, state. This motivates the following definition.

Definition 5.1. Let R be a relation scheme and let Σ be a set of FDs and MVDs which apply to it. The insertion of a tuple t into a relation $r(R)$, denoted by $(r, +t)$, is said to be a *valid insertion* if:

(i) $r \in \text{SAT}(\Sigma)$;

(ii) $t \notin r$;

(iii) $r \cup \{t\} \in \text{SAT}(\Sigma)$.

We illustrate this definition by the following example.

Example 5.1. Let $R = \{A, B, C\}$, $\Sigma = \{A \to B, B \to\to C\}$, $r = \{<1, 1, 1>, <1, 1, 0>$ and $r_1 = \{<1, 1, 1>, <1, 0, 0>\}$ (refer to the relations shown in Figure 5.2). Then $(r, +<0, 0, 1>)$, resulting in relation r_2, is a valid insertion, but not the insertions:

(a) $(r_1, +<0, 0, 1>)$, resulting in relation r_3, violates (i) since r_1 violates $A \to B$;

(b) $(r, +<1, 1, 0>)$, resulting in relation r_4, violates (ii);

(c) $(r, +<0, 1, 1>)$, resulting in relation r_5, violates (iii) since $B \to\to C$ is violated in r_5. □

r

A	B	C
1	1	1
1	1	0

insert <0, 0, 1> ⇒

r_2

A	B	C
1	1	1
1	1	0
0	**0**	**1**

r_1

A	B	C
1	1	1
1	0	0

insert <0, 0, 1> ⇒

r_3

A	B	C
1	1	1
1	0	0
0	**0**	**1**

r

A	B	C
1	1	1
1	1	0

insert <1, 1, 0> ⇒

r_4

A	B	C
1	1	1
1	1	0
1	**1**	**0**

r

A	B	C
1	1	1
1	1	0

insert <0, 0, 1> ⇒

r_5

A	B	C
1	1	1
1	1	0
0	**1**	**1**

Figure 5.2. Examples of valid and invalid insertions

We now define the notion of a set of attributes being unaffected by an insertion which, in the interpretation of Bernstein and Goodman [Bernstein and Goodman 1980], is the reason for update difficulties in a relation.

Definition 5.2. A set of attributes X is *unaffected* by a valid insertion $(r, +t)$ if $\pi_X(r) = \pi_X(r')$ where $r' = r \cup \{t\}$.

For example, let R and Σ be as defined in Example 5.1 and let $r = \{<1, 1, 1>\}$. Then BC is unaffected by the valid insertion $(r, +<0, 1, 1>)$ but is affected by the valid insertion $(r, +<1, 1, 0>)$. The relations resulting from these insertions are shown in Figure 5.3.

A	B	C
1	1	1

insert <0, 1, 1> ⇒

A	B	C
1	1	1
0	**1**	**1**

A	B	C
1	1	1

insert <1, 1, 0> ⇒

A	B	C
1	1	1
1	**1**	**0**

Figure 5.3. An example showing the affectation of attributes

As discussed previously in Chapter 3, only certain sets of attributes, called facts, are usually considered as semantically meaningful for the purpose of update. We adopt the same approach as in Chapter 3 and consider the following three possible sets of facts. The first is to define the set of facts to contain the attribute sets corresponding to the dependencies in the set Σ of MVDs and FDs derived from the database design. The second is to take into account the special symmetrical nature of an MVD constraint[8] and allow the set of attributes in any MVD $X \rightarrow\rightarrow R - XY$ corresponding to an MVD $X \rightarrow\rightarrow Y$ in Σ to also be a fact. The final possibility is to extend the set of facts still further and allow the set of attributes in any nontrivial FD or MVD implied by Σ to be fact. We now define unpredictable insertions for each of these possibilities.

Definition 5.3. Let R be a relation scheme and let Σ be a set of FDs and MVDs which apply to R. A valid insertion $(r(R), +t)$ is an *unpredictable insertion 1* (referred to subsequently as a UI_1) if there exists a nontrivial dependency $d \in \Sigma$ such that ATT(d) is unaffected by the insertion.

Definition 5.4. Define the set Σ' by $\Sigma' = \Sigma \cup \{X \rightarrow\rightarrow R - XY \mid X \rightarrow\rightarrow Y \in \Sigma\}$. A valid insertion $(r(R), +t)$ is an *unpredictable insertion 2* (referred to subsequently as a UI_2) if there exists a nontrivial dependency $d \in \Sigma'$ such that ATT(d) is unaffected by the insertion.

Definition 5.5. A valid insertion $(r(R), +t)$ is an *unpredictable insertion 3* (referred to subsequently as a UI_3) if there exists a nontrivial dependency $d \in \Sigma^+$ such that ATT(d) is unaffected by the insertion.

These definitions are illustrated by the following example.

Example 5.2. Let $R = \{A, B, C\}$, $\Sigma = \{A \rightarrow\rightarrow B\}$ and $r = <1, 1, 1>$. The insertion $(r, +<1, 0, 1>)$ is a UI_2 and a UI_3 since $A \rightarrow\rightarrow C$ is in both Σ' and Σ^+ and AC is unaffected by the insertion, but it is not a UI_1 since AB is affected by the insertion. □

---

[8]Refer to rule A4 in Section 2.4.

A	B	C
1	1	1

insert <1, 0, 1> ⇒

A	B	C
1	1	1
1	**0**	**1**

Figure 5.4. An example of a UI_2 and a UI_3

Both Bernstein and Goodman [Bernstein and Goodman 1980] and Vossen [Vossen 1988] also considered unpredictable deletions. According to their definitions, a relation r has an unpredictable deletion if the deletion of a tuple t from r results in a relation r' and both r and r' are in SAT(Σ) and the projections of r and r' onto the attributes of some dependency d in Σ are identical . However, it follows from this definition that if r has an unpredictable deletion when t is deleted from it then r' has an unpredictable insertion when t is inserted into it, and conversely if (r, +t) is an unpredictable insertion then the deletion of t from the resulting relation is an unpredictable deletion. So, since the occurrence of an unpredictable deletion always implies the existence of an unpredictable insertion and conversely the occurrence of an unpredictable insertion always implies the existence an unpredictable deletion, unpredictable deletions will not be considered.

Now, using these definitions of unpredictable insertions in relation instances, the corresponding normal forms which are free of these difficulties are defined.

Definition 5.6. A relation scheme R is said to be in *insertion normal form 1* (abbreviated subsequently to INF_1) if for every legal relation r defined over R, there doesn't exist a tuple t such that (r(R), +t) is a UI_1.

Definition 5.7. A relation scheme R is said to be in *insertion normal form* 2 (abbreviated subsequently to INF_2) if for every legal relation r defined over R, there doesn't exist a tuple t such that (r(R), +t) is a UI_2.

Definition 5.8. A relation scheme R is said to be in *insertion normal form* 3 (abbreviated subsequently to INF_3) if for every legal relation r defined over R, there doesn't exist a tuple t such that $(r(R), +t)$ is a UI_3.

We note that since $\Sigma \subseteq \Sigma' \subseteq \Sigma^+$, it follows directly from the definitions that $UI_1 \Rightarrow UI_2 \Rightarrow UI_3$ and so $INF_3 \Rightarrow INF_2 \Rightarrow INF_1$. Since INF_1 and INF_2 are defined in terms of the set of dependencies rather than its closure, a question that then arises is whether either of these normal form properties changes if Σ is replaced by an equivalent set of dependencies. We shall see in a later section that the INF_2 property does not change if Σ is replaced by an equivalent set, but the INF_2 property may change in the case of FD and MVD constraints but not in the case of MVD constraints alone.

5.3. THE CASE OF MVD CONSTRAINTS

5.3.1. The Relationship between Insertion Anomalies

We now consider the relationship between the different types of unpredictable insertions defined in the previous section. Firstly, the following result from Chapter 3 (Lemma 3.1) is recalled.

Lemma 5.1. *If R is a relation scheme and Σ is a set of MVDs and FDs that apply to R, then for any nontrivial dependency $X \rightarrow\rightarrow W$ or $X \rightarrow W$ in Σ^+ there exists a nontrivial dependency $X' \rightarrow\rightarrow Y$ or $X' \rightarrow Y$ in Σ such that $X' \subseteq X$.*

Using this result, it is now shown that UI_3 and UI_2 are equivalent.

Lemma 5.2. *An insertion $(r, +t)$ is a UI_3 if and only if it is also a UI_2.*

Proof.

<u>If</u>

Immediate since $\Sigma' \subseteq \Sigma^+$.

<u>Only If</u>

Suppose that a nontrivial MVD $X \to\to W$ in Σ^+ is unaffected by the insertion (r, $+t$). Since, by definition of a UI, $\pi_{XW}(r) = \pi_{XW}(r')$ where $r' = r \cup \{t\}$, there must exist a tuple $t' \in r$ such that $t[XW] = t'[XW]$. Also, by Lemma 5.1, there exists a nontrivial MVD in Σ of the form $X' \to\to Y$ where $X' \subseteq X$. Suppose to the contrary that (r, $+t$) is not a UI_2 and so both $X' \to\to Y$ and $X' \to\to Z$ where $Z = R - X'Y$ are affected by (r, $+t$). Partition R into the sets X', $Y \cap W$, $Y - W$, $Z \cap W$, $Z - W$ and denote the value of the tuple t on these sets by $<v_1, v_2, v_3, v_4, v_5>$ where $v_1 = t[X']$, $v_2 = t[Y \cap W]$, $v_3 = t[Y - W]$, $v_4 = t[Z \cap W]$ and $v_5 = t[Z - W]$. Because (r, $+t$) affects $X' \to\to Y$ and $X' \to\to Z$ but not $X \to\to W$, it follows that $t' = <v_1, v_2, v_3', v_4, v_5'>$ where $v_3 \neq v_3'$ and $v_5 \neq v_5'$. Since, from the definition of a UI3, $r \cup \{t\}$ satisfies $X' \to\to Y$, it follows that there has to be tuples t_1 and t_2 in $r \cup \{t\}$ with $t_1 = <v_1, v_2, v_3', v_4, v_5>$ and $t_2 = <v_1, v_2, v_3, v_4, v_5'>$. However, since $v_3 \neq v_3'$ and $v_5 \neq v_3'$ then t_1 and t_2 are distinct and neither can be equal to t and so both must be in r. Then, because r satisfies $X' \to\to Y$ and using t_1 and t_2, it follows that there is a tuple $<v_1, v_2, v_3, v_4, v_5> \in r$ which contradicts the definition of a UI_3 which requires that $t \notin r$. □

We consider now the relationship between a UI_1 and a UI_2. In contrast to the previous result, the following example shows that an insertion that is a UI_2 is not necessarily a UI_1.

Example 5.3. Let $R = \{A, B, C\}$, $\Sigma = \{A \to\to B\}$ and $r = \{<1, 1, 1>, <1, 2, 1>\}$. Then the insertion ($r$, $+<1, 3, 1>$) is a UI_2 since $A \to\to C \in \Sigma'$ and the insertion does not affect AC, but it is not a UI_1 since the insertion affects $A \to\to B$. In fact, there doesn't exist a tuple t such that (r, $+t$) is a UI_1 since any tuple t which is not a duplicate and doesn't affect AB can only be of the form $<1, 1, c_2>$ or $<1, 2, c_2>$ with $c_2 \neq 1$, and in both cases $r \cup \{t\}$ violates $A \rightarrow\rightarrow B$. However, there is another relation defined over the same relation scheme, namely $r = \{<1, 1, 1>\}$, such that $(r, +<1, 1, 2>)$ is a UI_1. It will be shown in the next section that this property holds in general, i.e. if $(r, +t)$ is a UI_2 then there exists r' and t' such that $(r', +t')$ is a UI_1. These insertions are shown in Figures 5.5 and 5.6. □

r

A	B	C
1	1	1
1	2	1

insert <1, 3, 1> ⇒

r_2

A	B	C
1	1	1
1	2	0
1	**3**	**1**

Figure 5.5. An insertion which affects attributes AB but not AC

r

A	B	C
1	1	1
1	2	1

insert <1, 1, c_2> ⇒

r_2

A	B	C
1	1	1
1	2	1
1	**1**	$\mathbf{c_2}$

r

A	B	C
1	1	1
1	2	1

insert <1, 2, c_2> ⇒

r_2

A	B	C
1	1	1
1	1	0
1	**2**	$\mathbf{c_2}$

Figure 5.6. Insertions which are not a UI_1

5.3.2. Insertion Normal Forms and 4NF

In this section the main results concerning the relationship between the insertion normal forms and 4NF when the constraints are restricted to MVDs will be derived. Firstly, it is shown that 4NF is equivalent to INF_3.

Theorem 5.3. *A relation scheme is in 4NF if and only if it is in INF_3.*

Proof.

Only If

We shall show the contrapositive that if R is not in INF_3 then it is not in 4NF. If R is not in INF_3 there exists a relation r and a tuple t such that r^*, where $r^* = r \cup \{t\}$, is in SAT(Σ) (and thus also in SAT(Σ_k)) and there is a tuple t' in r such that t[ATT(d)] = t'[ATT(d)] for a nontrivial dependency d in Σ^+. So two tuples in r^* are identical on the left-hand side of d and, since $r^* \in$ SAT(Σ_k), the left-hand side of d cannot be a superkey and so R is not in 4NF.

If Part.

We shall show the contrapositive that if R is not in 4NF then it is not in INF_3. We will do this by constructing a valid insertion that does not affect ATT(d) for a nontrivial MVD $d \in \Sigma^+$

As R is not in 4NF, there exists a nontrivial MVD $X \rightarrow\rightarrow Y \in \Sigma^+$ such that X is not a superkey. Since X and Y are assumed to be disjoint and $X^+ = X$ as Σ contains only MVDs, it follows that DEP(X) can be written as $\{X_1, \ldots, X_p, W_1, \ldots, W_n\}$ (refer to Section 2.4.2). From property (ii) of DEP(X), Y is equal to a union of elements of DEP(X). However, $Y \neq W_1 \ldots W_n$ since it would imply by property (i) of DEP that $R = XY$ and hence that $X \rightarrow\rightarrow Y$ is trivial. Since $W_1, \ldots, W_n$ are disjoint (property (ii) of DEP(X)), this implies that there exists a W_i such that $W_i \cap Y = \emptyset$. Then construct a relation r of two tuples t_1 and t_2 which are identical on every attribute except those in W_i.

It follows then from Lemma 5.24 that $X \rightarrow\rightarrow Y$ is unaffected by the valid insertion $(\{t_1\}, +t_2)$. □

A comment on the special nature of two tuple relations which has been used in this proof and in other contexts [Sagiv et al. 1981; Thalheim and Al-Fedhagi 1990] is appropriate at this point. If one starts instead with a relation of two or more tuples then, as was demonstrated in Example 5.3, a relation may not have an unpredictable insertion even if the relation scheme is not in 4NF. So the properties of two tuple relations is crucial in establishing this result. Next, it will be shown that INF_1, INF_2 and INF_3 are equivalent.

Theorem 5.4. *INF_1, INF_2 and INF_3 are all equivalent conditions on a relation scheme.*

Proof. Since it follows directly from the definitions of the insertion anomalies that $\text{INF}_3 \Rightarrow \text{INF}_2 \Rightarrow \text{INF}_1$, it suffices to show that $\text{INF}_1 \Rightarrow \text{INF}_3$ since it follows from Lemma 5.2 that INF_3 and INF_2 are equivalent.

Suppose to the contrary that R is in INF_1 but not in INF_3. By Theorem 5.3, R is not in 4NF and so, by Theorem 3.2, there exists a nontrivial MVD $X \rightarrow\rightarrow Y$ in Σ such that X is not a superkey. The same construction used in Theorem 5.3 then shows that there exists a valid insertion that does not affect XY and so one derives the contradiction that R is not in INF_1. □

One important corollary of the previous theorem is the following result which shows that INF_1 and INF_2 have the intuitively desirable property that they do not change if the set of MVDs is replaced by an equivalent set.

Corollary 5.5. *If Σ and Σ_1 are two equivalent set of MVDs, then a relation scheme is INF_1 (or INF_2) with respect to Σ if and only if it is INF_1 (or INF_2) with respect to Σ_1.*

Proof.

Follows from Theorem 5.4 and the fact that equivalent sets of dependencies have the same closure. □

5.4. THE CASE OF FD AND MVD CONSTRAINTS

5.4.1. The Relationship between Insertion Anomalies

In this section the results of the earlier sections are extended to the case where the set constraints contains both FDs and MVDs. Firstly, we show that Lemma 5.2 remains valid for this more general case.

Lemma 5.6. *An insertion $(r, +t)$ is a UI_3 iff it is also a UI_2.*

Proof.

If

Immediate from the fact that $\Sigma' \subseteq \Sigma^+$.

Only If

Denote by d the dependency in Σ^+ (either an FD or an MVD) which is unaffected by the insertion. From Lemma 5.1, there are only two cases to consider - the first is when the corresponding dependency in Σ is an FD and the other is when it is an MVD.

(i) The FD case.

Assume that the FD in Σ corresponding to d is $X \rightarrow W$. By definition of a UI_3, $\pi_{ATT(d)}(r) = \pi_{ATT(d)}(r')$ where $r' = r \cup \{t\}$. Then by Lemma 5.1, $X \subseteq ATT(d)$ and since, by definition of a UI_3, $r \cup \{t\}$ satisfies Σ it follows that $\pi_{XW}(r) = \pi_{XW}(r')$ and so the insertion is also a UI_2.

(ii) The MVD case.

Follows from Lemma 5.2. □

As was demonstrated earlier in Example 5.3, this equivalence result doesn't extend to UI_1 and UI_3.

5.4.2. Insertion Anomalies and 4NF

In this section the other main results of this chapter relating 4NF and the semantic normal forms INF_1, INF_2 and INF_3 for the case of FD and MVD constraints will be derived. Firstly, it is shown that Theorem 5.3 also extends to this case of FDs and MVDs.

Theorem 5.7. *A relation scheme is in 4NF if and only if it is in INF_3.*

Proof.

<u>Only If</u>

As for Theorem 5.3.

<u>If</u>

We shall show the contrapositive that if R is not in 4NF then it is not in INF_3. Since R is not in 4NF, there exists a nontrivial MVD $X \rightarrow\rightarrow Y$ in Σ^+ such that X is not a superkey. We now show that R is not in INF_3 by constructing a valid insertion that does not affect either XY or XZ where $Z = R - XY$.

As before, denote the elements of DEP(X) by $\{X_1, \ldots, X_p, X_1^+, \ldots, X_j^+, W_1, \ldots, W_n\}$. Since X is not a superkey and DEP(X) covers R, there is at least one element W_i in DEP(X) which is disjoint from X^+. By properties (ii) and (iii) of DEP(X), there are only two cases to consider: (a) there exists W_i such that $W_i \cap Y = \emptyset$, (b) $W_1 \ldots W_n \subseteq Y$.

For (a), construct a relation r of two tuples t_1 and t_2 such that the two tuples are identical on every attribute except for those in W_i. It follows then from Lemma 5.24 that XY is unaffected by the valid insertion $(\{t_1\}, +t_2)$.

For (b), consider the MVD $X \to\to Z$ where $Z = R - XY$. By inference rule A4, $X \to\to Z$ is in Σ^+ and is nontrivial since $X \to\to Y$ is nontrivial. Since $W_1 \ldots W_n \subseteq$ Y, it follows that there exists a W_i such that $Z \cap W_i = \emptyset$ and the same construction as in (a) shows a valid insertion that does not affect XZ. □

Next, we show that, as in the MVD case, INF_2 and INF_3 are equivalent.

Theorem 5.8. *A relation scheme is in INF_3 if and only if it is in INF_2.*

Proof.

Follows immediately from Lemma 5.6. □

A simple corollary of this result is the following lemma which shows that INF_2 has the desirable property of being invariant under replacement of the set of dependencies by an equivalent set.

Corollary 5.9. *If R is a relation scheme and Σ and Ψ are two equivalent set of FDs and MVDs which apply to R, then a relation scheme is INF_2 with respect to Σ if and only if it is INF_2 with respect with respect to Ψ.*

Proof. Follows directly from Theorem 5.8. □

Next, the following example shows that INF_1 is not equivalent to 4NF (and hence neither to INF_2 nor INF_3 by Theorems 5.7 and 5.8) when the constraints contain both

FDs and MVDs. This is in contrast to the case where the only constraints are MVDs, where INF_1 is equivalent to 4NF (Theorems 5.3. and 5.4)

Example 5.4. Consider the scheme $R = \{A, B, C\}$ and $\Sigma = \{A \rightarrow\rightarrow B, B \rightarrow A, B \rightarrow C\}$. We claim that R is in INF_1 but not in 4NF. It can be easily verified that the only candidate key in R is B and so R is not in 4NF since $A \rightarrow\rightarrow B$ violates the 4NF condition.

To see that R is in INF_1 suppose that there is a valid insertion $(r, +t)$ that doesn't affect a dependency $d \in \Sigma$. Then r must contain a tuple t' such that $t'[\mathrm{ATT}(d)] = t[\mathrm{ATT}(d)]$ and, since $r \cup \{t\} \in \mathrm{SAT}(\Sigma)$ and every dependency in Σ contains the candidate key B, this implies that $t' = t$ which contradicts property (ii) of a valid insertion. □

A syntactic characterisation of INF_1 is now provided by the following result.

Theorem 5.10. *If R is a relation scheme and Σ a set of FDs and MVDs which apply to R then the following are equivalent:*

(i) R is in INF_1;

(ii) For every nontrivial dependency $X \rightarrow\rightarrow Y$ or $X \rightarrow Y$ in Σ, XY is a superkey;

(iii) For every nontrivial dependency $X \rightarrow\rightarrow Y$ or $X \rightarrow Y$ in Σ, $X \rightarrow R - XY$ is also in S^+.

Proof.

(i) $\Rightarrow$ (ii)

Suppose to the contrary that there is a dependency such that XY is not a superkey. The same construction used in Theorem 5.7 then shows that there is a valid insertion which does not affect XY which contradicts the fact that R is in INF_1.

(ii) $\Rightarrow$ (iii)

If $X \rightarrow\rightarrow Y$ is a nontrivial MVD in Σ such that XY is a superkey then $XY \rightarrow R$ - XY is in Σ^+. Combining this with the fact that $X \rightarrow\rightarrow R$ - XY from rule A4 and using rule A9 shows that $X \rightarrow R$ - XY is also in Σ^+. Alternatively, if $X \rightarrow Y$ is a nontrivial FD in Σ such that XY is a superkey then X must also be a superkey, or else a simple application of the inference rules derives the contradiction that XY is not a superkey, and so $X \rightarrow R$ - XY is again in Σ^+.

(iii) $\Rightarrow$ (i)

If (iii) holds for any dependency $X \rightarrow Y$ or $X \rightarrow\rightarrow Y \in \Sigma$, then a simple application of the inference rules shows that XY must be a superkey. Suppose to the contrary that R is not in INF_1. Then by definition there exists a dependency $d \in \Sigma$ and relations r and $r' = r \cup \{t\}$ in SAT(Σ) such that r and r' have the same projection onto ATT(d). However this implies that r' has at least two tuples which are identical on ATT(d) and so ATT(d) cannot be a superkey which contradicts the fact that the set of attributes in every dependency in Σ is a superkey. □

As is illustrated by the following example, INF_1 has the rather undesirable feature that it may change when the set of dependencies is replaced by an equivalent set. This is in contrast to the case where the constraints contain only MVDs where it was shown to be invariant.

Example 5.5. Let $R = \{A, B, C\}$, $\Sigma = \{A \rightarrow\rightarrow B, B \rightarrow A, B \rightarrow C\}$ and $\Sigma_1 = \{A \rightarrow\rightarrow C, B \rightarrow A, B \rightarrow C\}$. It follows from the inference rules that B is the only candidate key and that Σ and Σ_1 are equivalent sets of dependencies. From Theorem 5.11, R is in INF_1 with respect to Σ because every dependency contains the candidate key B, but is not INF_1 with respect to Σ_1 because AC is not a superkey. □

5.5. CONCLUSIONS

In this chapter the justification for normalisation has been looked at from the perspective of avoiding *unpredictable insertions* to a relation. In this view, a relation scheme is defined as having an unpredictable insertion if different insertions into relations defined over the scheme require new values to be supplied for the attributes in an FD or MVD constraint in one insertion but not for the other.

The contribution of this chapter has been to extend this unpredictable update view of normalisation to the case of MVD constraints and then to FD and MVD constraints. Extending the approach for the FD case in an obvious manner, we consider the set of attributes in an FD or MVD constraint to be a fact and then define an unpredictable insertion to occur if different insertions into relations defined over the scheme require new information for a fact in one insertion but not in the other. Three equally plausible alternatives for the set of facts were then considered. The first being the attribute sets corresponding to the dependencies in Σ, the set of FD and MVD constraints supplied by the database design; the second being the attribute sets corresponding to the dependencies in Σ augmented by the MVDs $X \rightarrow\rightarrow R - XY$ corresponding to the MVDs $X \rightarrow\rightarrow Y$ in Σ; and the third being the attribute sets corresponding to the dependencies in the set of all nontrivial FDs and MVDs implied by Σ. Corresponding to each of these possibilities, a relation scheme is defined to be in one of the normal forms INF_1, INF_2 or INF_3 if the scheme does not have an unpredictable insertion with respect to the set of facts.

The main results derived in the chapter are as follows. For the case of MVD constraints, 4NF, INF_1, INF_2 and INF_3 were proved to be equivalent conditions on a relation scheme. In the more general case of FD and MVD constraints, it was shown that INF_2 and INF_3 are equivalent to each other and to 4NF, but INF_1 is a weaker condition than 4NF. A syntactic characterisation of INF_1 was also provided. A corollary of these results that the normal form INF_2 has the desirable property that it is invariant under replacement of Σ by an equivalent set but INF_1 does not possess the same property in general.

We now compare the approach taken in this chapter to the redundancy approach taken in Chapter 3. From the definition of an unpredictable insertion, if an insertion $(r, +t)$ is an unpredictable insertion then it follows that $r \cup \{t\}$ has at least two tuples which are identical on ATT(d), for some dependency d, and so the relation is redundant according to the definitions in Chapter 3. However, as the following example demonstrates, a relation being redundant does not imply that it has resulted from an unpredictable insertion because an unpredictable insertion has the more stringent requirement that the relation on which the insertion was performed is in SAT(Σ).

Example 5.6. Let $R = \{A, B, C\}$, $\Sigma = \{A \rightarrow\rightarrow B\}$ and consider the relation r shown in Figure 5.7. The relation r is in SAT(Σ) and contains two tuples which are identical on AB. However, r cannot have resulted from an unpredictable insertion since, if any tuple t is removed from r, the relation r - $\{t\}$ is not in SAT(Σ), and so the insertion (r - $\{t\}$, $+t$) is not an unpredictable insertion. □

A	B	C
1	1	1
1	1	0
1	0	1
1	0	0

Figure 5.7. A relation which is not a result of an unpredictable insertion

In contrast, in terms of relation schemes, the results of this chapter show that the insertion normal forms in the chapter are equivalent to the redundancy free properties defined earlier in Chapter 3. To be more specific, the following equivalences hold: R is in $\text{INF}_2 \Leftrightarrow R$ is not $\text{RED}_2 \Leftrightarrow R$ is in $\text{INF}_3 \Leftrightarrow R$ is not RED_3 (Theorems 3.14, 3.15, 5.7, 5.8) and R is in $\text{INF}_1 \Leftrightarrow R$ is not RED_1 (Theorems 3.17 and 5.11). Also, because of these equivalences and Theorem 3.18, it follows that the three insertion normal forms defined in this chapter, although they are not in general equivalent for the case of FD and MVD constraints, are equivalent if the MVDs are pure (refer to Chapter 2).

CHAPTER 6

FACT-BASED UPDATE ANOMALIES AND NORMAL FORMS

6.1. INTRODUCTION

In this chapter, the relationship between *fact-based update anomalies*[9] (as distinct from the key-based update anomalies analysed in Chapter 4) and the syntactic normal forms BCNF and 4NF is investigated. This approach is closest to the original justification of normal forms given by Codd [Codd 1972] and is also the informal justification often given for normal forms in database texts [Date 1990; El-Masri and Navathe 1989; Ullman 1988a]. Similar to the work on key-based update anomalies in Chapter 4, this approach is based on looking at desirable properties of a relation when it is updated, as opposed to the approach in Chapter 3 where the desirable static property of an absence of redundancy was investigated. However, similar to Chapter 3, the work in this chapter uses the fact-based approach which interprets subsets of attributes of a relation scheme, rather than the whole scheme, as being the fundamental units of information. Intuitively speaking, a fact-based update anomaly occurs when as a result of a relation scheme storing several independent facts, fact values in a relation cannot be independently manipulated without violating a relational constraint. Normalisation then attempts to overcome this problem by designing relations so that facts which cannot be independently updated are not stored

[9] To avoid repetition, in this chapter the term update anomaly will always refer to a fact-based update anomaly unless stated otherwise.

in the same relation scheme. We now illustrate these ideas with an example based on Codd's work.

Example 6.1. The set of attributes is $\{E\#, D\#, M\#\}$ and the set of FD constraints is $\{E\# \rightarrow D\#, D\# \rightarrow M\#, M\# \rightarrow D\#\}$ where the meanings of these attributes are as follows: *E#* - employee number, *D#* - department number, *M#* - employee number of manager. The primary key of the relation scheme is *E#*. The relation shown in Figure 6.1 satisfies the set of constraints.

E#	D#	M#
1	x	11
2	x	11
3	y	12
4	x	11
5	y	12
6	y	12
7	z	13

Figure 6.1. An example of update anomalies

Codd assumed that the facts correspond to FDs and so the set of facts is $\{E\#\ D\#, D\#\ M\#\}$. Codd allowed relations to contain null values but required that the primary key of every tuple in a relation must be full (contains no null values). The relation shown above then has the following processing difficulties. Firstly, one cannot insert a tuple consisting of a fact value about a department and its manager, such as department *w* has manager *15*, into the relation since this would imply that in the new tuple the value of the primary key, *E#*, would be null. So a new fact value relating a department and its manager cannot be independently inserted into the relation. This problem is referred to as an *insertion anomaly*. We note that implicit in this approach is the assumption that containing null values are permitted in tuples. This differs from the work in Chapter 5 where it was assumed that partial tuples are not permitted in relations.

The second problem is that if one wants to delete the fact value that employee *7* leaves the organisation then, if this is done by deleting the tuple for employee *7*, the fact value that department *z* has a manager *13* is also deleted because it is the only tuple in the relation containing this information. This difficulty is referred to as a *deletion anomaly*.

The last type of processing anomaly occurs when the manager of a department changes. For example, if the manager of department *x* is to be changed to *14* then, in order to ensure that the updated relation will satisfy the set of constraints, several tuples will have to be changed and in general the number of tuples to be changed will vary with time. This problem is referred to as a *replacement anomaly*. Looked at from another perspective (which will be used later in the formal definitions), this difficulty occurs when the replacement of a fact value in only a single tuple, rather than all the tuples storing the same fact value, causes the constraints to be violated. □

As mentioned in Chapter 1, Codd didn't formally investigate update anomalies and it was not until relatively recently that a formal analysis was undertaken by Chan [Chan 1989]. Chan provided formal definitions of the three types of fact-based update anomalies - *insertion anomaly, deletion anomaly* and *replacement anomaly* - and investigated their relationship to syntactic normal forms, both for single and multiple relation schemes, in the case of FD constraints. Even though we feel that some aspects of the relationship between insertion and deletion anomalies and normal forms require further investigation, we don't pursue the issue in this chapter. This is because, as will be discussed in more detail later in Section 6.6, we feel that a more thorough investigation is outside the scope of this thesis since it requires the incorporation of null values and their effect on dependency satisfaction. Instead, the contribution of this chapter is to investigate the relationship between a replacement anomaly and the normal forms 4NF and BCNF from a different perspective from that of Chan.

Chan defined a replacement anomaly as occurring when the replacement of the values of some attributes in a tuple results in the constraints being violated. The attributes whose values in a relation can be modified are an arbitrary, user defined set and not necessarily related to the FD constraints. Also, replacements which violate key-uniqueness are permitted. We propose an alternative definition which we feel is more consistent with Codd's original fact-based approach and the basic principles of the relational model. No explicit limitation is placed on which attribute-values can be changed in a tuple as long as the attributes are members of a single fact, but only replacements which satisfy key-uniqueness are permitted. We then define a relation scheme as having a replacement anomaly if there exists a legal relation defined over the scheme such that the replacement of a fact value in a tuple results in key-uniqueness being maintained but the FD and MVD constraints being violated; in other words, a fact value in a tuple cannot be independently replaced. We now illustrate this definition by an example.

Example 6.2. Let $R = \{A, B, C\}$, $\Sigma = \{A \rightarrow B, B \rightarrow C\}$ and let the set of facts be $\{AB, BC\}$. Consider the relation shown in Figure 6.2. R has a replacement anomaly since when tuple <*2, 2, 1*> is replaced by <*2*, ***1***, ***2***>, so that a fact value for BC is replaced, the new relation satisfies the key constraints (since A is the only candidate key) but violates $B \rightarrow C$. □

A	B	C
1	1	1
2	2	1

replace <2, 2, 1> by <2, **1**, **2**>

⇓

A	B	C
1	1	1
2	**1**	**2**

Figure 6.2. An example of a replacement anomaly

As we noted in Chapter 3, there are several approaches to determining the set of facts in a relation scheme. In this chapter, as in Chapters 3 and 5, the original approach of Codd is adopted and we assume that the set of facts contains the sets of attributes appearing in the FDs and MVDs which apply to a relation scheme[10]. We note that the alternative approach, which was discussed in more detail in Chapter 3, is to assume that the set of facts can be defined independently of the FD and MVD constraints [Beeri et al. 1981; Chan 1989; Desai et al. 1986; Maier et al. 1987; Maier et al. 1983; Maier et al. 1986; Maier and Ullman 1983; Sciore 1980]. In accordance with our approach in Chapters 3 and 5, we allow for three different sets of facts which correspond to different alternatives for the set of constraints. These alternative sets of constraints are: the set Σ of FDs and MVDs derived from the database design, Σ plus all MVDs $X \rightarrow\rightarrow R - XY$

[10]This is a slight extension of Codd's approach since he considered only FDs and so assumed that facts corresponded to the attribute sets in the FD constraints, whereas we have extended this approach and assumed the set of attributes in an MVD to also be a fact.

corresponding to the MVDs $X \rightarrow\rightarrow Y$ in Σ; and lastly, the set of all nontrivial FDs and MVDs implied by Σ. According to each of these choices for the set of facts, different subtypes of a replacement anomaly are defined along with three normal forms for relation schemes, referred to as $FRNF_1$, $FRNF_2$ and $FRNF_3$ respectively, which are free of the corresponding type of replacement anomaly.

The main results derived in this chapter on the relationship between syntactic normal forms and the replacement normal forms are as follows. In the case of FD constraints, FRNF_1, FRNF_2 and FRNF_3 are proven to be equivalent to each other and to BCNF. A similar result is also shown to hold for the case where the only dependencies are MVDs when BCNF is replaced by 4NF. However, in contrast to what occurred for the redundancy and unpredictable insertion properties examined in Chapters 3 and 5, for the most general case where the set of constraints includes both FDs and MVDs, the normal forms FRNF_1, FRNF_2 and FRNF_3 are again equivalent to each other and to 4NF. Thus the normal form FRNF_1 is not equivalent to the properties INF_1 and not RED_1 (defined in Chapters 3 and 5) which are similarly defined in terms of only the dependencies in Σ.

The other contribution of this chapter is to define several subtypes of a more restrictive type of replacement anomaly and investigate the relationship between their absence in a relation scheme and the syntactic normal forms BCNF and 4NF. Motivated by the works of Biskup and the *entity-relationship approach* to data modelling [Batini et al. 1991; Biskup 1989; Biskup and Dublish 1991; Chen 1976], rather than regarding the set of attributes in a dependency as an indivisible unit of information, the approach adopted is to view the sets of attributes X and Y in a dependency $X \rightarrow Y$ or $X \rightarrow\rightarrow Y$ as playing different roles. X is interpreted as representing some entity and the attributes in Y are interpreted as the properties of X. We then consider fact replacements where the Y-values can change, but not the X-values which, in this approach, represent the identity of an entity. A modification anomaly is then defined to be a replacement anomaly where the only attribute values which can be modified are those belonging to the right-hand side of a dependency. As for a replacement anomaly, we define three semantic normal forms (referred to as $FMNF_1$, $FMNF_2$ and $FMNF_3$) corresponding to the same three choices for the set of facts. The main results derived in this chapter concerning these normal forms is to show that they are equivalent to the fact replacement normal forms discussed earlier.

6.2. THE DEFINITIONS OF UPDATE ANOMALIES AND FACT-BASED NORMAL FORMS

The formal definitions of fact-based modification and replacement anomalies are now presented. As mentioned in Section 6.1, the definitions are based on the widely used approach of interpreting sets of attributes in a relation scheme as representing the fundamental units of information for retrieval and update. [Bernstein 1976; Chan 1989; Desai et al. 1987; Hall et al. 1976; Jajodia and Ng 1983; Nijssen and Halpin 1989; Sciore 1980; Vossen 1988].

6.2.1. Modification Anomalies

The interpretation of a fact-based modification anomaly is based on differentiating between the roles of the attribute set X and the attribute set Y in a fact corresponding to an FD $X \rightarrow Y$ or an MVD $X \rightarrow\rightarrow Y$. The set of attributes X is interpreted as representing the attributes of an entity and the set of attributes Y as the properties of the entity X. Before giving the formal definition of the modification anomalies, we recall the definition of a compatible tuple given in Chapter 4.

Definition 6.1. Let R be a relation scheme, Σ a set of FDs and MVDs which apply to R and $r(R)$ a relation defined over R. A tuple t^* is said to be *compatible* with r if $r \cup \{t^*\}$ is a relation in SAT(Σ_K).

Using this definition and modelling the modification of a tuple as the deletion of the tuple followed by the insertion of the new tuple, a modification violation is now defined.

Definition 6.2. A relation $r(R)$ has a *modification violation$_1$* (abbreviated subsequently to *MV_1*) with respect to a *reduced set* Σ of FDs and MVDs if there exists a tuple $t \in r$ and a tuple t^* defined over R such that:

(i) $r \in \mathrm{SAT}(\Sigma)$;

(ii) t^* is compatible with $(r - \{t\})$;

(iii) t and t^* differ only on some set of attributes Y' such that there exists either $X \to Y$ or $X \to\to Y$ in Σ with $Y' \subseteq Y$;

(iv) $\{r - \{t\}\} \cup \{t^*\}$ violates Σ**.**

We note that a reduced set of dependencies is used in this definition because the dependencies in such a set cannot have their left or right hand sides decomposed and so are irreducible units of information [Desai et al. 1986; Desai et al. 1987]. We now use the definition of a modification violation in a relation instance to define a normal form for relation schemes which ensures that these violations can never occur.

Definition 6.3. A relation scheme R has a *modification anomaly 1* (abbreviated subsequently to *MA_1*) if there exists a relation $r(R)$ which has an MV_1. A relation scheme R is in *fact modification normal form$_1$* (abbreviated subsequently to *$FMNF_1$*) if it doesn't have an MA_1.

As was done in previous chapters, we now extend these definitions to allow for different sets of facts.

Definition 6.4. A relation $r(R)$ has a *modification violation 2* (abbreviated subsequently to *MV_2*) if it satisfies all the conditions of Definition 6.2 except that condition (iii) is changed to:

(iii) t and t^* differ only on some set of attributes Y' such that there exists either $X \rightarrow Y$ or $X \rightarrow\rightarrow Y$ in Σ' with $Y' \subseteq Y$;

Definition 6.5. A relation scheme R has a *modification anomaly 2* (abbreviated subsequently to MA_2) if there exists a relation $r(R)$ which has an MV_2. A relation scheme R is in *fact modification normal form 2* (abbreviated subsequently to $FMNF_2$) if it doesn't have an MA_2.

We note that in the case where the set of constraints contains only FDs, an MV_1 and an MV_2 are identical and thus so are $FMNF_1$ and $FMNF_2$.

Definition 6.6. A relation $r(R)$ has a *modification violation 3* (abbreviated subsequently to MV_3) if it satisfies all the conditions of Definition 6.2 except that condition (iii) is changed to:

(iii) t and t^* differ only on some set of attributes Y' such that there exists a nontrivial dependency $X \rightarrow Y$ or $X \rightarrow\rightarrow Y$ in Σ^+ with $Y' \subseteq Y$

Definition 6.7. A relation scheme R has a *modification anomaly 3* (abbreviated subsequently to MA_3) if there exists a relation $r(R)$ which has an MV_3. A relation scheme R is in *fact modification normal form$_3$* (abbreviated subsequently to $FMNF_3$) if it doesn't have an MA_3.

The following example illustrates the previous definitions.

Example 6.3. Let R = {EMP, $DEPT$, $MNGR$} and Σ = {$EMP \rightarrow DEPT$, $DEPT \rightarrow MNGR$}. The only candidate key is EMP. The relation r shown in Figure 6.3 has both an MV_1 and an MV_3 when the tuple <*Ball, Accounts, Jones*> is updated to <*Ball, **Finance**, Jones*>, resulting in the relation r', since r and r' differ only on $DEPT$, which is the right-hand side of $EMP \rightarrow DEPT$, and r' satisfies Σ_K but violates $DEPT \rightarrow MNGR$. □

r

EMP	DEPT	MNGR
Jones	Tax	Allen
Carter	Finance	Stark
Ball	Accounts	Jones

replace <Ball, Accounts, Jones> by <Ball, **Finance**, Jones>

⇓

r'

EMP	DEPT	MNGR
Jones	Physics	Allen
Carter	Finance	Stark
Ball	**Finance**	Jones

Figure 6.3. An example illustrating modification violations

We note that since $\Sigma \subseteq \Sigma' \subseteq \Sigma^+$, the following implications follow directly from the definitions: a relation r has an $MV_1 \Rightarrow r$ has an $MV_2 \Rightarrow r$ has an MV_3, and thus, correspondingly in relation schemes: a relation scheme R is in $FMNF_3 \Rightarrow R$ is in $FMNF_2$ $\Rightarrow R$ is in $FMNF_1$. We also note that in the definitions of modification violations, the dependency which is violated by the modification can be any dependency in Σ, it is not necessarily the dependency whose right-hand side attribute values are modified by the update that is violated. For instance, in Example 6.3, the attribute on the right-hand side of $EMP \rightarrow DEPT$ is modified but the FD violated is $DEPT \rightarrow MNGR$.

A few comments on the relationships between the above definitions and the processing problems defined in the previous chapters are appropriate at this point. For the redundancy properties introduced in Chapter 3 (refer to Definitions 3.1 - 3.3), it is easy to prove that if a relation has two or more tuples which are identical on the set of attributes in a dependency then, by changing the values of the right-hand side attributes in one of the tuples to values which are distinct from any appearing in the relation, a modification violation results. This implies that the following implications are valid for relation schemes: $FMNF_1 \Rightarrow \neg RED_1$, $FMNF_2 \Rightarrow \neg RED_2$, $FMNF_3 \Rightarrow \neg RED_3$. In terms of relation instances firstly, the converse property - that if a relation has a modification violation then it must contain redundancy - is not valid since, in Example 6.3, a relation was constructed which has a modification violation but does not contain redundancy. However, as will be seen later, in terms of relation schemes the converses do hold and the fact modification normal forms defined above will be demonstrated to be equivalent to the corresponding redundancy-free properties.

There are several differences between fact-based modification anomalies and the key-based modification anomalies analysed in Chapter 4. Firstly, in the fact-based definitions, only attribute-values in the right-hand side of a dependency can be modified, whereas in the key-based modification anomalies either no restriction is placed on which attributes can be modified (in the case of a key-based MV_1) or any attributes which are not part of the primary key (in the case of a key-based MV_2) or part of any candidate key (in the case of a key-based MV_3) are allowed to be modified. Secondly, key-based modifications require that the identity of a tuple be preserved during the modification. These differences mean that the definitions are not comparable and reflect the fact that they make different interpretations regarding the semantics of a relation. In the key-based approach, a tuple is regarded as the fundamental semantic unit of information whose identity is given by its candidate key values, whereas in the fact-based definitions it is assumed that the set of attributes in a constraint is the atomic unit of information whose identity in a tuple is given by the values of the attributes.

Lastly, in relation to the unpredictable insertions investigated in Chapter 5, it was noted in Section 5.5 that if a relation has an unpredictable insertion then it contains redundancy and so, from the previous discussion concerning redundancy and fact-based modification anomalies, it follows that the relation must also have a modification violation. The following example shows, however, that a relation may have a modification violation without having an unpredictable insertion.

Example 6.4. Let $R = \{A, B, C\}$ and let $\Sigma = \{A \rightarrow\rightarrow B\}$. Consider the relation r shown in Figure 6.4. The relation r has an MV_1 when the tuple $<1, 0, 0>$ is replaced by $<1, \mathbf{2}, 0>$. We claim though that there is no tuple t such that $(r, +t)$ is an unpredictable insertion of any type. This is because the definitions of all the types of unpredictable insertions require that $r \cup \{t\}$ be in SAT(Σ) and $t[A] = 1$, but there is no tuple with this property. To verify this assertion, because of the tuples already in r, either $t[B]$ or $t[C]$ must be distinct from both 0 and 1 in order for t not to be a duplicate of a tuple already in r. It then follows easily from the definition of an MVD that $r \cup \{t\}$ violates $A \rightarrow\rightarrow B$ and so $(r, +t)$ is not a valid insertion. □

A	B	C
1	0	0
1	1	1
1	1	0
1	0	1

Figure 6.4. A relation without an unpredictable insertion

In terms of relation schemes though, it will be seen later that the fact modification normal forms are equivalent to the corresponding unpredictable insertion normal forms.

6.2.2. Replacement Anomalies

In this section we define replacement violations, replacement anomalies and the corresponding normal forms which are free of the anomalies. As discussed in Section 6.1, our definition of a replacement anomaly is based on interpreting the set of attributes in a dependency as being an indivisible unit of information and thus the values of any attributes in a fact can be replaced. So the modification anomalies defined in the previous section are special cases of replacement anomalies where only some of the attributes in a fact are allowed to be modified. Formal definitions are now presented.

Definition 6.8. A relation $r(R)$ has a *replacement violation 1* (abbreviated subsequently to RV_1) if it satisfies all the conditions of Definition 6.2 except that condition (iii) is changed to:

(iii) The set of attributes on which t and t^* differ is a subset of ATT(d) where d is an FD or MVD in Σ.

Definition 6.9. A relation scheme R has a *replacement anomaly 1* (abbreviated subsequently to RA_1) if there exists a relation $r(R)$ which has an RV_1. A relation scheme R is in *fact replacement normal form 1* (abbreviated subsequently to $FRNF_1$) if it doesn't have an RA_1.

Definition 6.10. A relation $r(R)$ has a *replacement violation 2* (abbreviated subsequently to RV_2) if it satisfies all the conditions of Definition 6.2 except that condition (iii) is changed to:

(iii) The set of attributes on which t and t^* differ is a subset of ATT(d) where d is an FD or MVD in Σ'.

Definition 6.11. A relation scheme R has a *replacement anomaly 2* (abbreviated subsequently to RA_2) if there exists a relation $r(R)$ which has an RV_2. A relation scheme R is in *fact replacement normal form 2* (abbreviated subsequently to $FRNF_2$) if it doesn't have an RA_2.

As in the case of a modification anomaly, for the situation where the only constraints are FDs we note that RV_1 and RV_2 are equivalent and thus so are $FRNF_1$ and $FRNF_2$.

Definition 6.12. A relation $r(R)$ has a *replacement violation 3* (abbreviated subsequently to RV_3) if it satisfies all the conditions of Definition 6.2 except that condition (iii) is changed to:

(iii) The set of attributes on which t and t^* differ is a subset of ATT(d) where d is a nontrivial FD or MVD in Σ^+.

Definition 6.13. A relation scheme R has a *replacement anomaly 3* (abbreviated subsequently to RA_3) if there exists a relation $r(R)$ which has an RV_3. A relation scheme R is in *fact replacement normal form 3* (abbreviated subsequently to $FRNF_3$) if it doesn't have an RA_3.

The following example illustrates the previous definitions.

Example 6.5. Let $R = \{A, B, C\}$, $\Sigma = \{A \rightarrow B, B \rightarrow C\}$ and let $r = \{t_1, t_2\}$ where $t_1 = <1, 1, 1>$ and $t_2 = <2, 2, 1>$. The relation r is shown in Figure 6.5. Then r has both an RV_1 and an RV_3 when $<2, 2, 1>$ is updated to $<2, \mathbf{1}, \mathbf{2}>$ since the attributes BC are contained in the FD $B \rightarrow C$ and the resulting relation, r', is in SAT(Σ_k) (since A is the only candidate key) but violates the FD $B \rightarrow C$. We also claim that r has no MV_1 and so RV_1 and MV_1 are not equivalent conditions on a relation instance. To verify this, consider the modification of either t_1 or t_2. From the definition of an MV_1, only the right-hand side of an FD can change and so only the B value or the C value (but not both) of a tuple can be changed. Suppose firstly that t_1 is modified to t'. If $t_1[B]$ is changed then $A \rightarrow B$ is still satisfied because $t'[A] \neq t_2[\mathrm{A}]$ and $B \rightarrow C$ is still satisfied because $t'[C] = t_2[C]$. If instead $t_1[C]$ is modified, then AB is still satisfied because $t'[A] \neq t_2[A]$ and BC is also satisfied because $t'[B] \neq t_2[B]$. Similar arguments apply if t_2 is modified and so r does not have an MV_1. □

r

A	B	C
1	1	1
2	2	1

replace <2, 2, 1> by <2, **1**,**2**>

⇓

r'

A	B	C
1	1	1
2	**1**	**2**

Figure 6.5. An example illustrating replacement violations

We note that it follows directly from the definitions of replacement violations that the following relationships hold between the different types of violations and anomalies: a relation r has an $\mathrm{RV}_1 \Rightarrow r$ has an $\mathrm{RV}_2 \Rightarrow r$ has an RV_3, and thus in relation schemes: a relation scheme R is in $\mathrm{FRNF}_3 \Rightarrow R$ is in $\mathrm{FRNF}_2 \Rightarrow R$ is in FRNF_1. Also, the following relationships hold between the replacement violations defined in this section and the modification violations defined in the previous section: $MV_1 \Rightarrow RV_1$, $MV_2 \Rightarrow RV_2$, $MV_3 \Rightarrow RV_3$, and correspondingly between the normal forms: $FRNF_1 \Rightarrow FMNF_1$, $FRNF_2 \Rightarrow FMNF_2$, $FRNF_3 \Rightarrow FMNF_3$. It is noted that, as shown in Example 6.5 where a relation was constructed that had an RV_1 and an RV_3 but neither an MV_1 nor an MV_3, some of the converses to these implications are not valid.

6.3. THE CASE OF FD CONSTRAINTS

In this section, syntactic normal forms will be derived which are equivalent to the semantic modification and replacement normal forms defined in the previous sections for the case where the only constraints are FDs.

6.3.1. Modification Anomalies and Normal Forms

In this section we derive the main results concerning the relationship between BCNF and the modification normal forms. Firstly, two preliminary lemma derived in Chapter 4 (Lemmas 4.10 and 4.5) are recalled.

Lemma 6.1. *Let Σ be a set of FDs and let $X \rightarrow A$ an FD in Σ such that X is not a superkey. Then for every candidate key K, $K - XA \neq \emptyset$.*

Lemma 6.2. *If r(R) be a relation in SAT(Σ) and X is a set of attributes that is not a superkey, then for any tuple $t \in r$ there exists a tuple t' such that $t[X^+] = t'[X^+]$, $t'[A] \notin r[A]$ for all $A \in (R - X^+)$ and $r \cup \{t'\}$ is a relation in SAT(Σ).*

The next lemma is also a simple consequence of a result in an earlier chapter (Corollary 3.3).

Lemma 6.3. *Let Σ be a reduced set of FDs. If $X \rightarrow A$ is a nontrivial FD in Σ^+ such that X is not a superkey, then there exists an FD $Z \rightarrow A$ in Σ such that Z is not a superkey.*

Some preliminary results will now be derived concerning the relationship between the different types of modification violations. These results are both interesting in their own right as well as being needed later in the proofs of the results concerning normal forms. The first lemma shows the equivalence of an MV_1 and an MV_3.

Lemma 6.4. *A relation r has an MV_1 if and only if r also has an MV_3.*

Proof.

Only If

Automatic since $\Sigma \subseteq \Sigma^+$.

If

Let A be the attribute updated and let $X \rightarrow A$ the corresponding nontrivial FD in Σ^+. By Lemma 6.3 there exists $Z \rightarrow A$ in Σ and so it follows immediately from the definition of an MV_1 that the update is also an MV_1. □

Consequences of the previous result are the following important corollaries which establish the intuitively desirable results that the property of a relation having an MV_1 (and thus also a relation scheme being in $FMNF_1$) is unchanged if the set of dependencies is replaced by an equivalent set.

Corollary 6.5. *A relation r has an MV_1 with respect to a set of FDs iff it also has an MV_1 with respect to any equivalent set of FDs.*

Proof. Follows directly from Lemma 6.4 and the fact that equivalent sets of FDs have the same closure. □

Corollary 6.6. *A relation scheme R is in $FMNF_1$ with respect to a set of FDs if and only if it is $FMNF_1$ with respect to any equivalent set of FDs.*

Proof. Immediate from Corollary 6.5. □

Next, the main results of this section are derived.

Lemma 6.7. *Let R be a relation scheme, Σ a set of FDs which apply to R and suppose that R is not in BCNF. For every nonempty relation r(R) in SAT(Σ), there exists a tuple t' such that $r \cup \{t'\}$ has both an MV_1 and an MV_3.*

Proof. Since R is not in BCNF, by Lemma 6.3 there is a nontrivial FD $Z \to A$ in Σ such that Z is not a superkey. Hence by Lemma 6.2, for any tuple $t \in r$ there exists a tuple t' such that $t[Z^+] = t'[Z^+]$, $t'[B] \notin r[B]$ for all $B \in (R - Z^+)$ and $r' = r \cup \{t'\}$ is a relation in SAT(Σ). Define the tuple t^* by $t^*[R - A] = t'[R - A]$ and let $t^*[A]$ be assigned a value such that $t^*[A] \notin r[A]$. The claim is that r' has an MV_1 and an MV_3 when t' is modified to t^*.

Conditions (i), (iii), and (iv) of the definition of an MV_1 are automatically satisfied. For the compatibility condition (ii), if $A \notin K$, then $t^*[K] = t'[K]$ and so (ii) is satisfied because r' is in SAT(Σ) and hence also in Σ_K. If $A \in K$, then $t^*[K] \notin r'[K]$ by the construction of t^* and so (ii) is again satisfied. Hence r' has an MV_1 and also an MV_3. □

Although the previous lemma shows that for any relation r defined on a relation scheme which is not in BCNF, there exists t' such that $r \cup \{t'\}$ has an MV_1, in Example

6.5 it was shown that r itself may not have an MV_1 (and hence neither an MV_3 by Lemma 6.4)

We now present the main theorem of this section.

Theorem 6.8. *If R is a relation scheme, then the following are equivalent:*
(i) R is in BCNF;
(ii) R is in $FMNF_3$;
(iii) R is in $FMNF_1$.

Proof. We shall show (i) $\Rightarrow$ (ii) $\Rightarrow$ (iii) $\Rightarrow$ (i).

(i) $\Rightarrow$ (ii)

Assume to the contrary that R is in BCNF but is not in $FMNF_3$. Then there exists tuples t, t^* and a relation $r(R)$ such that the relation r^*, where $r^* = \{r - \{t\}\} \cup \{t^*\}$, is in $SAT(\Sigma_K)$ but violates some FD $X \rightarrow A$ in Σ. If $X \rightarrow A$ is violated in r^*, there have to be at least two tuples with identical X values. But this implies that X cannot be a superkey because $r^* \in SAT(\Sigma_K)$ and so the assumption that R is in BCNF is contradicted.

(ii) $\Rightarrow$ (iii)

Follows directly from the definition of the normal forms.

(iii) $\Rightarrow$ (i)

We show the contrapositive that if R is not in BCNF then it is not in $FMNF_1$. Firstly, since attribute domains are assumed to be infinite and any relation that contains no duplicate attribute values automatically satisfies Σ, it follows that there are an infinite number of relations which satisfy Σ. Let r be any such nonempty relation. It follows from Lemma 6.7 that there exists t such that $r \cup \{t\}$ has an MV_1 and so R is not in $FMNF_1$. □

It is noted that the last part of proof in the above theorem is also a consequence of Theorem 3.7 and the observation, mentioned earlier in Section 6.2.1, that if a relation scheme is in $FMNF_1$ then it is not RED_1 (refer to Definition 3.1 in Section 3.1).

6.3.2. Replacement Anomalies and Normal Forms

In this section, the relationship between BCNF and the replacement normal forms is investigated. Firstly, a preliminary result is presented.

Lemma 6.9. *Let R be a relation scheme and let Σ be a set of FDs which apply to R. If R is not in BCNF then every relation r(R) that is in SAT(Σ) and contains at least two tuples has an* RV_3.

Proof. Since R is not in BCNF, then by definition there exists a nontrivial FD $X \rightarrow A$ in Σ^+ such that X is not a superkey. A simple application of inference rules then shows that $R - A \rightarrow A$ is nontrivial FD in Σ^+ and so, by definition of a replacement violation, every attribute in R can be modified during a replacement. Let t_1 and t_2 be any two tuples in r. Define the tuple t^* by: $t^*[X] = t_1[X]$, $t^*[R - X] \notin r$. Such a tuple t^* always exists because of the assumption of infinite domains. The claim is that r has an RV_3 when t_2 is replaced by t^*.

Conditions (i), (iii) and (iv) of the definition of an RV_3 are automatically satisfied by the definition of t^*. For condition (ii), it follows from Lemma 6.1 that for any candidate key K, $K - X \neq \emptyset$ and so by definition of t^*, $t^*[K - X] \notin r[K - X]$ and so (ii) is also satisfied. □

The following example demonstrates that a relation defined over a scheme which is not in BCNF may not have an RV_1 and so, in contrast to the situation for MV_1 and MV_3 where they were shown in Lemma 6.4 to be equivalent conditions on a relation, RV_3 and RV_1 are not equivalent conditions on a relation.

Example 6.6. Let $R = \{A, B, C\}$ and let $\Sigma = \{A \rightarrow B\}$. The only key is AC and so R is not in BCNF. Let r be the relation shown in Figure 6.6. We claim that r has no RV_1. Suppose that the first tuple, called t_1, is changed. If r has an RV_1 when t_1 is replaced by t^*, then since $A \rightarrow B$ is the only FD, the update must violate this FD and so $t^*[A] = t_2[A]$. But by definition of an RV_1, t_1 and t^* can differ only on AB and hence $t^*[AC] = t_1[AC] = t_2[AC]$ which violates the key uniqueness condition. Similar arguments apply if the other tuple is changed and so r doesn't have an RV_1. □

A	B	C
1	1	1
2	2	1

Figure 6.6. A relation which has an RV_3 but not an RV_1

Although the previous example showed that a relation r may have an RV_3 but not an RV_1, the next result shows that there always exists a tuple t' such that $r \cup \{t'\}$ has both an RV_1 and an RV_3.

Lemma 6.10. *Let R be a relation scheme, Σ a set of FDs which apply to R. If R is not in BCNF then, for every nonempty relation r(R) in SAT(Σ), there exists a tuple t' such that $r \cup \{t'\}$ has both an RV_1 and an RV_3.*

Proof. Immediate from Lemma 6.7 and the fact that an MV_1 is also an RV_1 and an RV_1 is also an RV_3. □

Also, in addition to the fact that a relation may have an RV_3 but not an RV_1, the following example shows that a relation may have an RV_1 but not an MV_1 (and hence neither an MV_2 by Lemma 6.7).

Example 6.7. As in the previous example, let $R = \{A, B, C\}$ and let $\Sigma = \{A \rightarrow B\}$. The only key is AC and so R is not in BCNF. Let r be the relation shown in Figure 6.7. r has an RV_1 when <2, 2, 2> is replaced by <**1**, 2, 2>. However, r does not have an MV_1 since modifying the value of B in either of the two tuples does not cause $A \rightarrow B$ to be violated. □

A	B	C
1	1	1
2	2	2

replace <2, 2, 2> by <**1**, 2, 2>

$\Downarrow$

A	B	C
1	1	1
1	2	2

Figure 6.7. A relation with an RV_1 but not an MV_1

However, despite the different types of modification and replacement violations in relation instances not being equivalent, the next theorem shows that an absence of any of the replacement anomalies in a relation scheme are equivalent conditions to each other and to BCNF.

Theorem 6.11. *If R is a relation scheme, then the following are equivalent:*

(i) R is in BCNF;

(ii) R is in $FRNF_1$;

(iii) R is in $FRNF_3$.

Proof. As for the proof of Theorem 6.8 and noting that $FRNF_3 \Rightarrow FMNF_3$. □

6.4. THE CASE OF MVD CONSTRAINTS

In this section, the relationship between the semantic normal forms and 4NF is examined for the case where the only constraints are MVDs.

6.4.1. Modification Anomalies and Normal Forms

In this section we look at extending the results of Section 6.3.1 to the case of MVD constraints. As we now indicate, several of the lemmas established in that section are not valid for this case. Firstly, Lemma 6.2 is not valid since it was shown in Example 6.4 that, for a relation r in SAT(Σ), there doesn't always exist a tuple t such that $r \cup \{t\}$ is also in SAT(Σ). Next, the following counter-example demonstrates that Lemma 6.3 is not valid in the MVD case.

Example 6.8. Let $R = \{A, B, C\}$ and let $\Sigma = \{A \rightarrow\rightarrow B\}$. It follows from inference rule A4 (see Chapter 2) that $A \rightarrow\rightarrow C$ is a nontrivial MVD in Σ^+ but there is no MVD in Σ of the form $X \rightarrow\rightarrow C$. □

Similarly, the following counter-example shows that even if we weaken Lemma 6.3 by requiring that the corresponding dependency be in Σ', rather than in Σ as required by the lemma, then the modified lemma is still not valid for the MVD case.

Example 6.9. Let $R = \{A, B, C, D\}$ and let $\Sigma = \{A \rightarrow\rightarrow BC, A \rightarrow\rightarrow CD\}$. By applying rule A4, $\Sigma' = \{A \rightarrow\rightarrow BC, A \rightarrow\rightarrow CD, A \rightarrow\rightarrow D, A \rightarrow\rightarrow B\}$. It follows from a simple application of the inference rules that $A \rightarrow\rightarrow C$ is a nontrivial

MVD in Σ^+ yet there is no MVD in Σ' of the form $X \rightarrow\rightarrow C$.

□

However, the following result shows that a weaker result than Lemma 6.3 holds for the MVD case.

Lemma 6.12. *If $X \rightarrow\rightarrow AY$ is a nontrivial MVD in Σ^+ such that X and AY are disjoint then there exists a nontrivial MVD $V \rightarrow\rightarrow AW$ in Σ'.*

Proof. From Lemma 4.1, there exists a nontrivial MVD $X' \rightarrow\rightarrow Y' \in \Sigma$ with $X' \subseteq X$. A is disjoint from X' since A is disjoint from X, and so A must either be a member of Y' or Z' where $Z' = R - XY$. The result follows immediately since $X \rightarrow\rightarrow Z \in \Sigma$ from inference rule A4. □

We now explore the relationships between the different types of modification violations. Firstly, the following example shows that a relation may have an MV_3 but not an MV_2 and so Lemma 6.4 does not extend to the MVD case.

Example 6.10. Let $R = \{A, B, C, D, E\}$ and let $\Sigma = \{AB \rightarrow\rightarrow C, CD \rightarrow\rightarrow A\}$. It follows from an application of the inference rules that $\Sigma' = \{AB \rightarrow\rightarrow C, AB \rightarrow\rightarrow DE, CD \rightarrow\rightarrow A, CD \rightarrow\rightarrow BE\}$ and $ABD \rightarrow\rightarrow CD$ is a nontrivial MVD in Σ^+. Consider the relation r shown in Figure 6.8. It can easily be seen that r has an MV_3 when <*0, 0, 0, 0, 0*> is replaced <*0, 0*, ***1***, ***1***, *0*> since only the value of CD, which is the right-hand side of an MVD in Σ^+, is modified by the replacement and the modified relation violates $CD \rightarrow\rightarrow A$. However, r does not have an MV_2 since only the attribute values of one of the attribute sets $\{C, DE, A, BE\}$ can be modified and no change to these values in either of the tuples results in an MV_2. □

r

A	B	C	D	E
1	1	1	1	1
0	0	0	0	0

replace <0, 0, 0, 0, 0> by <0, 0, **1**, **1**, 0>

$\Downarrow$

r'

A	B	C	D	E
1	1	1	1	1
0	0	**1**	**1**	0

Figure 6.8. A relation with an MV_3 but not an MV_2

Next, we show that a relation may have an MV_2 but not an MV_1.

Example 6.11. As in Example 6.8, let $R = \{A, B, C\}$ and let $\Sigma = \{A \rightarrow\rightarrow B\}$. Consider the relation r shown in Figure 6.9. r has an MV_2 when the tuple <*1, 0, 1*> in r is replaced by the tuple <*1, 0,* ***0***> since the MVD $A \rightarrow\rightarrow C$ is in Σ^+ and the resulting relation, r', violates $A \rightarrow\rightarrow B$. However, r does not have an MV_1 since no change to the B value of a tuple in r can result in $A \rightarrow\rightarrow B$ being violated. □

r

A	B	C
1	1	1
1	0	1

replace <1, 0, 1> by <1, 0, **0**>

$\Downarrow$

r'

A	B	C
1	1	1
1	0	**0**

Figure 6.9. A relation with an MV_2 but not an MV_1

This example also demonstrates that Corollary 6.5 is not valid for the MVD case; in other words, a relation can have an MV_1 with respect to one set of MVDs but not with respect to an equivalent set. This is because the relation given in the example was shown not to have an MV_1 with respect to the set $\{A \rightarrow\rightarrow B\}$, but the argument used also shows that the relation has an MV_1 with respect to the equivalent set $\{A \rightarrow\rightarrow C\}$. However, as will be seen later, Corollary 6.6 is still valid for the MVD case. In addition, Example 6.11 shows that Lemma 6.7 is not valid in the MVD case since it can easily be verified that there is no tuple t which can be inserted into the relation so that $r \cup \{t\}$ has an MV_1 because, for this to happen, one must have that $t[A] = 1$ and then it can easily be verified that $r \cup \{t\} \notin \text{SAT}(\Sigma)$ and so does not have an MV_1.

Despite many of the preliminary lemmas of Section 6.3.1 not being valid for the MVD case, we now show that the main theorem of that section, Theorem 6.8 on the relationship between the BCNF and the modification normal forms, also extends to the MVD case.

Theorem 6.13. *If R is a relation scheme, then the following are equivalent:*

(i) R is in 4NF;

(ii) R is in $FMNF_3$*;*

(iii) R is in $FMNF_2$*;*

(iv) R is in $FMNF_1$*.*

Proof. We shall show (i) ⇒ (ii) ⇒ (iii) ⇒ (iv) ⇒ (i).

(i) ⇒ (ii)

As for Theorem 6.8.

(ii) ⇒ (iii)

Follows from the definitions of the normal forms.

(iii) ⇒ (iv)

Follows from the definitions of the normal forms.

(iv) ⇒ (i)

As mentioned earlier in Section 6.2.1, if a relation r is in SAT(Σ) and contains two or more tuples that are identical on the set of attributes of an MVD $X \rightarrow\rightarrow Y$ in Σ, then it is easy to verify that the relation has an MV_1 when the Y-value of one of the tuples is changed to a value which is distinct from the Y-values in r. Taking the converse, this implies that if a scheme is in $FMNF_1$ then it is not RED_1 and so by Theorems 3.10 and 3.11 it is in 4NF. □

6.4.2. Replacement Anomalies and Normal Forms

In this section, we consider the extension of the results of Section 6.3.2 for the case of MVD constraints. Firstly, we show that the analogue of Lemma 6.9 is still valid in the this case.

Lemma 6.14. *Let R be a relation scheme and let Σ be a set of MVDs which apply to R. If R is not in 4NF then every relation r(R) in SAT(Σ) that contains at least two tuples has an* RV_3.

Proof. Since R is not in 4NF, then by definition there exists a nontrivial MVD $X \rightarrow\rightarrow Y$ in Σ^+ such that X is not a superkey. A simple application of inference rules then shows that $X\ (R - XY) \rightarrow\rightarrow Y$ is nontrivial MVD in Σ^+ and so, by definition of a replacement violation, every attribute in R can be modified during a replacement. The same argument used in Lemma 6.9 shows that there exists a relation that has an RV_3. □

With regard to the equivalence of the different types of replacement violations in relations, a relation may have an RV_3 but not an RV_2 and a relation may have an RV_2 but not an RV_1. These conclusions follow from the results of Section 6.3.2 and the fact that if a relation has a modification violation then it also has a replacement violation of the same type. Also, Lemma 6.10 is not valid for the MVD case because it was shown in Example 6.4 that, for any relation r, there doesn't always exist a tuple t such that $r \cup \{t\}$ satisfies a set of MVD constraints and so, by definition of a replacement violation, $r \cup \{t\}$ does not have a replacement violation of any type. However, we now show that, as for modification anomalies, the analogue of Theorem 6.11 is valid for the MVD case.

Theorem 6.15. *If R is a relation scheme then the following are equivalent:*

(i) R is in 4NF;

(ii) R is in $FRNF_3$;

(iii) R is in $FRNF_2$;

(iv) R is in $FRNF_1$.

Proof. (i) ⇒ (ii) follows from the same argument used in the first part of Theorem 6.13. (ii) ⇒ (iii) and (iii) ⇒ (iv) follow directly from the definitions of update anomalies and (iv) ⇒ (i) follows from Theorem 6.13 and the fact that $FRNF_1 \Rightarrow FMNF_1$. □

6.5. THE CASE OF FD AND MVD CONSTRAINTS

In this section, we investigate the relationship between the various types of replacement normal forms and the syntactic normal forms for the most general case where both FDs and MVDs are present.

6.5.1. Modification Anomalies and Normal Forms

We firstly prove that Theorem 6.13 extends to the case of FD and MVD constraints.

Theorem 6.16. *The following are equivalent conditions on a relation scheme R:*

(i) R is in 4NF;

(ii) R is in $FMNF_3$;

(iii) R is in $FMNF_2$;

(iv) R is in $FMNF_1$.

Proof.

(i) ⇒ (ii)

As for Theorem 6.8.

(ii) ⇒ (iii)

Follows directly from the definitions of the normal forms.

(iii) $\Rightarrow$ (iv)

Follows directly from the definitions of the normal forms.

(iv) $\Rightarrow$ (i)

We shall prove the contrapositive that if R is not in 4NF then it is not in FMNF_1. If R is not in 4NF, then by Theorem 3.2 there exists a nontrivial dependency $X \rightarrow Y$ or $X \rightarrow\rightarrow Y$ in Σ such that X is not a superkey. We now show how to construct a relation r which has an MV_1 which implies that R is not in FMNF_1.

As discussed in Section 2.4.2, the dependency basis of X, DEP(X), can be written as $\{X_1, \ldots, X_p, X_1^+, \ldots, X_j^+, W_1, \ldots, W_n\}$. Since DEP($X$) covers R and X is not a superkey, then at least one of the W's must be nonempty.

Suppose firstly that the dependency in Σ violating 4NF is the FD $X \rightarrow Y$. Construct a relation of two tuples in which the tuples are identical on all attributes except one of the W's. It follows from Lemma 4.24 that the relation is in SAT(Σ). Set the value of Y in one of the tuples to a distinct value such that the tuples are now different on Y. It can be easily verified that this update satisfies the requirements of an MV_1 and so R is not in FMNF_1.

Alternatively, suppose that the dependency in Σ violating4NF is the MVD $X \rightarrow\rightarrow Y$. We now consider separately the cases where: (a) XY is a superkey and (b) XY is not a superkey.

(a) XY is not a superkey.

Form the tableau T_{XY} as described in Section 2.5, let $T^* = chase_\Sigma(T_{XY})$ and let $r = \rho(T^*)$ be the relation derived from any one-to-one valuation ρ. From Lemma 3.6, r consists of two or more tuples and every tuple is identical on XY and thus every tuple in r is distinct on Z where $Z = R - XY$. By changing the Y-value of any tuple in r to a new value, it follows that r has an MV_1.

(b) XY is a superkey.

Form the tableau T_X as described in Section 2.5, let $T^* = chase_\Sigma(T_X)$ and let $r = \rho(T^*)$ be the relation derived from any one-to-one valuation ρ. By Lemma 3.6, all tuples in r are identical on X and r contains at least two tuples. Suppose firstly that there exist at least two tuples in r which don't have identical Z-values. In this case, change the Y-value of one of the tuples to a new value that is not in r. It follows that the new relation is in SAT(Σ_k) but violates $X \rightarrow\rightarrow Y$ and so r has an MV$_1$. Alternatively, suppose that every tuple in r is identical on Z. From Lemma 2.2, this implies that $X \rightarrow Z$ is an FD in Σ^+ and it is nontrivial since X and Z are disjoint. From a straightforward application of Lemma 2.2 ([Beeri and Vardi 1981b]), there has to be a nontrivial FD $V \rightarrow A$ in Σ for every attribute A in Z. Modify the A-value in any tuple of r to a new value. It follows that the new relation is in SAT(Σ_k) but violates $V \rightarrow A$ and so r has an MV$_1$. □

We note that the equivalence between FMNF$_1$ and 4NF demonstrated in this theorem is in contrast to the results obtained in Chapters 3 and 5 for the semantic properties similarly defined using the attribute sets in Σ as facts. It was shown in Chapter 3 (Example 3.1) that the property of a relation scheme not being RED$_1$ is a weaker condition than 4NF and in Chapter 5 it was similarly shown (Example 5.5) that the semantic normal form INF$_1$ is a weaker condition than 4NF. It is now demonstrated that the differences between these semantic normal forms disappear when the set of dependencies is pure (refer to Section 2.4.3).

Lemma 6.17. *If R is a relation scheme and Σ is a pure set of FDs and MVDs that apply to R then R is in FMNF$_1$ iff it is not RED$_1$.*

Proof.

Only If

As for Theorem 6.13.

If

The contrapositive, that if a scheme is not in $FMNF_1$ then it is in RED_1, will be shown. Using Theorem 3.17, it is sufficient to show that there is a dependency $X \rightarrow Y$ or $X \rightarrow\rightarrow Y$ in Σ such that XY is not a superkey. We recall that if R is not in $FMNF_1$, there exists a relation r in SAT(Σ) such that if a tuple t in r is modified to a tuple t' by changing the values of Y', then $Y' \subseteq Y$ for some dependency $X \rightarrow Y$ or $X \rightarrow\rightarrow Y$ in Σ and the new relation, r', violates Σ but satisfies Σ_K. Thus in r', either an FD $V \rightarrow W$ or an MVD $V \rightarrow\rightarrow W$ is violated and, for this to happen, there has to be at least two tuples in r' which are identical on V and, since r' satisfies Σ_K, V cannot be a superkey. In the case of the dependency violated being the FD $V \rightarrow W$, a simple application of the inference rules shows that VW cannot be a superkey. Alternatively, if the dependency violated is the MVD $V \rightarrow\rightarrow W$, then VW cannot be a superkey or else Lemma 3.12 would imply that $V \rightarrow R - VW$ is in Σ^+ contradicting the assumption that Σ is pure. □

Since INF_1 was shown in Chapter 5 to be equivalent to $\neg RED_1$, it follows from this lemma that INF_1 is also equivalent to $FMNF_1$ when the set of dependencies is pure.

6.5.2. Replacement Anomalies and Normal Forms

In this section, we investigate the relationship between the replacement normal forms and 4NF. We prove that a similar result to Theorem 6.16 is valid for replacement normal forms.

Theorem 6.18. *If R is a relation scheme then the following are equivalent:*

(i) R is in 4NF;

(ii) R is in $FRNF_3$*;*

(iii) R is in $FRNF_2$*;*

(iv) R is in $FRNF_1$*.*

Proof.

(i) ⇒ (ii)

As for Theorem 6.8.

(ii) ⇒ (iii)

Follows directly from the definitions of the normal forms.

(iii) ⇒ (iv)

Follows directly from the definitions of the normal forms.

(iv) ⇒ (i)

Follows from Theorem 6.16 and the fact that $FRNF_1 \Rightarrow FMNF_1$. □

6.6. RELATED WORK AND DISCUSSION

The work of Desai *et al.* [Desai et al. 1986; Desai et al. 1987], while not directly related to normalisation, gives an interesting perspective on the fact-based approach to interpreting database semantics. They proposed, as we have assumed in this chapter, that a tuple in a relation is not the atomic unit of information. For instance, taking an example from their work, given the relations scheme *R* = {*Student, Course, Grade*} and a tuple <*Smith, Comp352, A*>, then the tuple contains more than the fact value that *Smith* has a grade of *A* in *Comp352*. It also contains the following subfacts values:

Smith is a student,

COMP352 is a course,

A is a grade,

Smith has taken COMP352,

Smith has an A,

Somebody has an A in Comp352, and

Smith has an A.

Since some facts are subsets of others, they consider the facts in a relation scheme to have a lattice structure based on set inclusion. They then defined procedures for defining the insertion and deletion of fact values in a relation. In general, these procedures may result in more than one tuple being modified since they allow for a fact value to appear in several tuples and also permit null values. This is contrast to the work in this chapter where only a single tuple is modified during an update and null values are not allowed. The main result of their first paper was to show that if one defined a view mapping by extracting all the distinct values of a fact in a relation, then the view mapping has several desirable properties [Dayal and Bernstein 1982; Furtado and Casanova 1985] with respect to insertions and deletions. We note that in this paper, dependencies were not considered and so the procedures for inserting or deleting fact values do not involve checking for dependency violations.

In the second paper, they considered the presence of FD constraints and assumed that the set of attributes in an FD is always a fact. The procedures for the insertion and deletion of facts were then modified to ensure that the updated relation satisfies the set of FD constraints. It was then noted that it some cases it is not possible to delete a fact and satisfy a set of FD constraints and they derived an algorithm for testing if a fact is deletable. In contrast again to the work in this chapter, key constraints were not explicitly considered.

The work by Chan [Chan 1989] is more closely related to our approach than the work of Desai *et al.* just discussed. Chan investigated the relationship between the three types of update anomalies and normal forms in the case of FD constraints. He initially adopted a general approach and allowed the set of facts to be independent of the FD constraints and, based on Codd's original assumption that the primary key values in a table not be null, he defined a relation scheme to be free of insertion and deletion anomalies if every fact contains the primary key of the relation scheme R. His rationale for this definition is that if this condition is satisfied, then the insertion of a fact value results in the satisfaction of the constraint that the primary key be non null and similarly, in the case of a deletion, the deletion of a tuple containing a fact can only result in the removal of fact values that include the primary key. Chan then adopted a more restricted approach to defining the set of facts and proved that if the set of facts is chosen to correspond to the nontrivial FDs in Σ^+, then a relation scheme is free of insertion and deletion anomalies if and only if the scheme is in BCNF. Although Chan's approach implicitly assumes that nulls are permitted in relations, no account was taken of the effect of null values on the definitions of FD satisfaction, key values or normal forms. We feel that a more complete investigation of the relationship between normal forms and fact-based insertion and deletion anomalies requires the incorporation of null values and so, as mentioned in the introduction to the chapter, we have not pursued the matter as it is outside the scope of the thesis. Also, in the case of deletions, deleting a tuple in order to delete a fact value is only one possible approach and, as noted by others [Desai et al. 1986; Desai et al. 1987; Fagin et al. 1986; Fagin et al. 1983], other approaches are possible since defining precisely what is meant by a fact deletion is a rather subtle issue. We noted in Section 6.1 that Chan also investigated the relationship between a replacement anomaly and normal forms but, as discussed previously, his definitions differ from ours and so his results are not comparable to ours.

6.7. CONCLUSIONS

In this chapter, we have investigated the relationship between the absence of several types of fact-based replacement anomalies in a relation scheme and the syntactic normal forms 4NF and BCNF. We defined a relation scheme to have a replacement anomaly if there exists a legal relation defined over the scheme such that the replacement of a fact value in a tuple results in key uniqueness being satisfied but the set of dependencies being violated. As in Chapters 3 and 5, we assumed that the set of facts contains the sets of attributes appearing in the FD and MVD constraints. Three possible alternatives for the set of constraints were then considered. The first being the set Σ of FD and MVD constraints supplied by the database design, the second being the set Σ augmented by the MVDs $X \rightarrow\rightarrow R - XY$ corresponding to the MVDs $X \rightarrow\rightarrow Y$ in Σ, and the third being the set of all MVDs logically implied by Σ. Corresponding to each of these possible sets of facts, a relation scheme is defined to be in the normal forms $FRNF_1$, $FRNF_2$ or $FRNF_3$ if the scheme does not have a replacement anomaly with respect to the set of facts.

The other contribution of the chapter has been to investigate the relationship between a more restricted type of replacement anomaly, called a modification anomaly, and the normal forms BCNF and 4NF. The definition of a modification anomaly is based on distinguishing between the roles played between the sets of attributes X and Y in the FD $X \rightarrow Y$ or MVD $X \rightarrow\rightarrow Y$. X is interpreted as representing an entity and the attributes in Y as being the properties of the entity. A modification anomaly is then defined as a special type of replacement anomaly which occurs when only the Y-values in a tuple can be changed. As for a replacement anomaly, three normal forms for a relation scheme were defined which guarantee an absence of a modification anomaly in the scheme according to which set of constraints is chosen.

The main results derived in the chapter are as follows. For the case where the only constraints are FDs, the modification normal forms $FMNF_1$, $FMNF_2$, $FMNF_3$ and the replacement normal forms $FRNF_1$, $FRNF_2$ and $FRNF_3$ were shown to be equivalent to each other and to BCNF. For the cases where the set of constraints contains only MVDs or, most generally, where the set of constraints contains both FDs and MVDs, the modification normal forms $FMNF_1$, $FMNF_2$, $FMNF_3$ and the replacement normal forms $FRNF_1$, $FRNF_2$ and $FRNF_3$ were shown to be equivalent to each other and to 4NF. This equivalence between $FMNF_1$, $FRNF_1$ and 4NF in the case where there are both FD and MVD constraints is in contrast to the results of Chapters 3 and 5 where the semantic normal forms, which were similarly defined using only the dependencies in Σ, were shown to be weaker conditions than 4NF.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1. CONCLUSIONS

In this thesis, we have addressed the problem of providing a formal justification for the use of normal forms in relational database design. We have formally defined four different properties that it is desirable that a relation scheme should possess. These properties are: an absence of redundancy, an absence of key-based update anomalies, an absence of unpredictable insertions and an absence of fact-based update anomalies. The motivation for each of these properties is as follows. An absence of redundancy ensures that a relation will not contain duplicate information; an absence of key-based update anomalies ensures that as long as a simple type of constraint, called a key constraint, is satisfied by a relation after an update then the new relation will automatically satisfy all the more complex types of constraints (such as FD and MVDs); an absence of unpredictable updates ensures that in making insertions to a relation, different insertions do not require different information to be supplied; and lastly, an absence of fact-based update anomalies ensures that the atomic units of information stored in a relation can be independently updated.

For each of the properties just mentioned and for three possible choices for the set of dependencies, normal forms were defined (called semantic normal forms) which encapsulate the desirable processing properties with respect to the set of dependencies. The problem of deriving equivalent conditions (called syntactic normal forms) to these semantic normal forms, in terms of the properties of the set of dependencies, was then addressed. For some of the semantic normal forms, we proved that, depending on the types of constraints permitted, either BCNF or 4NF are equivalent conditions to the semantic normal forms, but for other semantic normal forms the equivalent syntactic normal forms are weaker than BCNF or 4NF. In particular, we proved that if the set of dependencies includes both FDs and MVDs but not the symmetric counterparts of MVDs, then the semantic normal forms defined with respect to this set of dependencies are weaker than 4NF. For another desirable property of a relation scheme, an absence of a key-based modification anomaly in which no candidate key value is changed, we proved that in the case of the only constraints being FDs, the equivalent syntactic normal form is a new normal form which lies between 3NF and BCNF and, in the case of both FD and MVD constraints, the equivalent syntactic normal form is a weaker condition than 4NF. An overview of the relationships between the various semantic normal forms defined in this thesis and the syntactic normal forms BCNF and 4NF is presented in Figures 7.1 - 7.3 for the cases where: the only constraints are FDs, the only constraints are MVDs, and lastly, when the constraints include both FDs and MVDs (RED_1, RED_2 and RED_3 are defined in Chapter 3; MA_1, MA_2, MA_3 and PANF are defined in Chapter 4; INF_1, INF_2 and INF_3 are defined in Chapter 5; $FMNF_1$, $FMNF_2$, $FMNF_3$, $FRNF_1$, $FRNF_2$, and $FRNF_3$ are defined in Chapter 6).

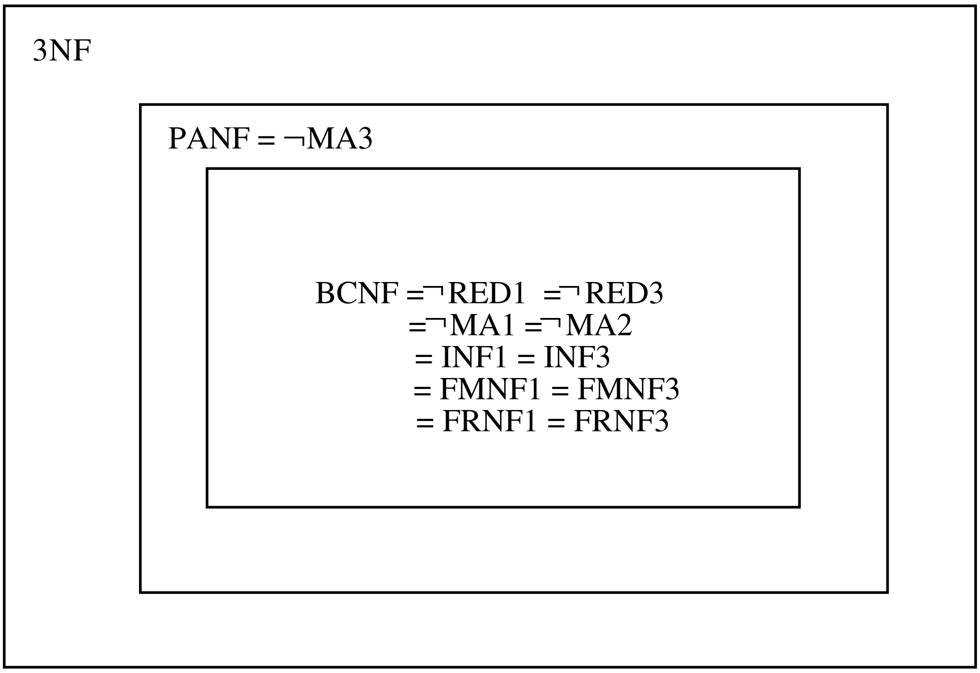

Figure 7.1. The relationship between normal forms for the FD case

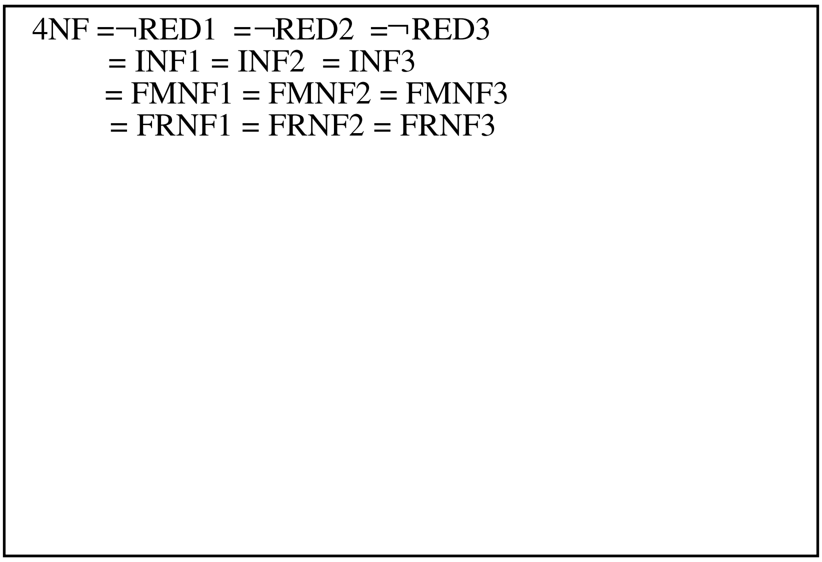

Figure 7.2. The relationship between normal forms for the MVD case

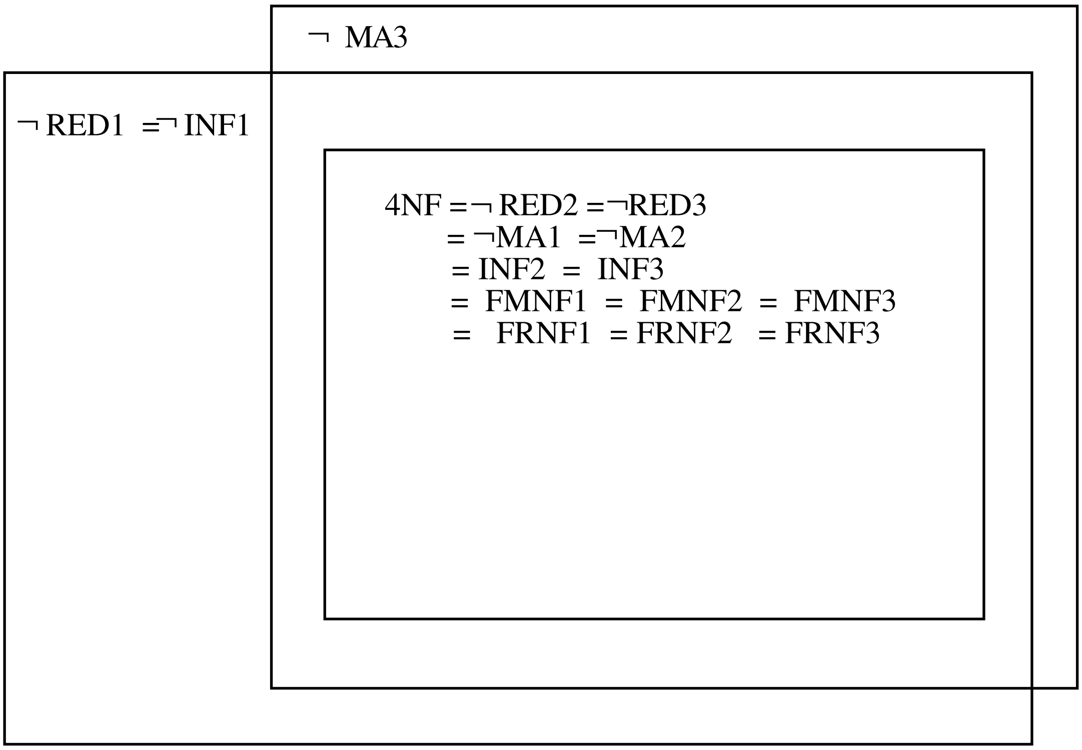

Figure 7.3. The relationship between normal forms for the case of FDs and MVDs

7.2. FUTURE WORK

The approach taken in this thesis also suggests several other areas of research where a similar approach would be useful. These areas include: normal forms in the presence of more general constraints than FDs and MVDs, generalisation of the model to allow nulls in relation, normalisation in the presence of multiple relations, normal forms for some of the newer data models such as the nested relational model and deductive data models. We now discuss each of these extensions in turn.

7.2.1. Normal Forms and Join Dependencies

As discussed in Chapter 2, several normal forms have been defined, both in the research literature and well known database texts [Date 1990; El-Masri and Navathe 1989; Yang 1986], for the case where the constraints can also include *join dependencies* (*JDs*). However, it has been shown [Orlowska and Zhang 1992] that many of these normal forms are not equivalent and it is not clear what their properties are. There is some justification for the original normal form of *PJ/NF* defined by Fagin since its definition is semantic in nature and implies that a relation scheme being in PJ/NF is equivalent to the condition that the relation scheme has no key-based insertion anomaly. Most of the other normal forms, though, are defined syntactically. For instance, Maier [Maier 1983] defined a relation scheme to be in *weak 5NF* if every component of the nontrivial JDs implied by the set of constraints is a superkey and noted that this condition is weaker than PJ/NF. An interesting area of research would be to extend the definitions of the semantically desirable properties introduced in this thesis to the case of JDs and investigate if any of the definitions of normal forms for this case can be derived from some desirable semantic property.

7.2.2. Normal Forms and Null Values

Another natural extension of the work in this thesis is to generalise the type of values allowed in a relation and permit the presence of *null values* [Codd 1979; Codd 1986; Codd 1987]. We noted in Chapter 6 that a more thorough examination of the relationship between fact-based insertion and deletion anomalies and normal forms requires null values to be taken into account since this approach implicitly assumes that null values are permitted. From a purely practical point of view, it is also desirable to investigate the implications of permitting null values since incomplete information occurs frequently in practical database applications.

There are at least three interpretations for null values that are commonly used [Atzeni and DeAntonellis 1993] (although some people have argued that there are more [Date 1990]). These interpretations are: to represent a value that is known to exist but whose current value is unknown (referred to as an *existential null*); to represent a value that is not applicable (often referred to as an *inapplicable null*); and lastly, to represent a value for which nothing, neither its existence or applicability, is known (referred to as a *no-information null*). While most of the research on nulls has concentrated on their effect on query interpretation, some researchers have also investigated the changes required to the definitions of dependency satisfaction and the corresponding dependency inference rules when null values are permitted [Atzeni and Morfuni 1984; Lien 1979; Vassilou 1979; Vassilou 1980; Zaniolo 1984]. As would be expected, one effect on the inference rules is that the rules are weaker than in the case where nulls are not permitted and, in particular, the transitive rule (see rule A3 in Chapter 2) is no longer valid. Another topic of research would be to investigate whether the results obtained in this thesis remain valid when null values in a relation are permitted and the definitions of normal forms correspondingly altered.

7.2.3. Multiple Relations

In this thesis, normalisation has been studied only in the context of a single relation. A natural generalisation is to consider the justification for normal forms in the presence of multiple relations. This adds a level of complexity because, in the presence of multiple relations, constraints may span relations and so one firstly has to consider precisely what it means for a set of relations to satisfy a set of constraints. This issue has been extensively investigated in the context of *universal relation databases* [Korth et al. 1984; Maier and Ullman 1983; Maier et al. 1984; Ullman 1983b] and the common consensus today on what it means for a set of relations to satisfy a set of constraints is based on the *weak instance* approach first formulated in the early 1980's by Honeyman and Sagiv [Honeyman 1980; Sagiv 1981]. Roughly speaking, a set of relations is defined to satisfy a set of constraints if there exists a single relation, called the universal relation, containing every attribute in the set of relation schemes such that the universal relation satisfies the set of constraints and each relation is contained in the projection of the universal relation onto the set of attributes in the relation[11].

---

[11] We note though that while this is a relatively natural definition of the satisfaction of constraints in the FD case, it is not entirely satisfactory in the case of MVD constraints because, in this case, every set

The question of whether some sets of relation schemes have more desirable properties has been researched by several people. The earliest was by Sagiv [Sagiv 1983] who investigated the property of *independence* which ensures that the satisfaction of the constraints in each relation is sufficient to imply satisfaction by all the relations in the weak instance sense. Since then, many other desirable properties have been proposed [Atzeni and Chan 1987; Chan and Atzeni 1992; Chan and Hernandez 1991; Chan and Hernàndez 1988a; Chan and Hernàndez 1988b; Chan and Mendelzon 1987; Graham and Yannakis 1984; Hernàndez and Chan 1991; Sagiv 1988] but, apart from the work of Chan [Chan 1989], the issue of normalisation in the context of multiple relations has not been investigated. As mentioned in Chapter 7, the approach of Chan, although utilising the fact-based approach which has been widely used in this thesis, differs from our approach and so another avenue of research would be to investigate the extension of our approach to multiple relations.

7.2.4. Normal Forms in other Data Models

One natural generalisation of the relational model which has been extensively investigated in the last decade is the *nested relational model* (also referred to as *non first normal form* or NF^2) [Jaeschke and Schek 1982; Makinouchi 1977; Schek and Scholl 1986]. The nested relational data model relaxes the rule that attribute values in a relation be atomic and instead allows a set of values. For some applications where the data has a naturally nested structure, modelling the application data as nested relations can result in a more natural model of the application. Even for applications implemented as flat relations, being able to group attribute values and apply set operations is still often required and such queries can be formulated more naturally in a nested query language [Levene and

---

of relations has a weak instance by adding the appropriate number of tuples. For MVDs a different definition of satisfaction (called completeness) has been proposed [Graham et al. 1986].

Loizou 1989; Roth et al. 1987; Roth et al. 1988] than in a flat query language, such as SQL [Date 1987], where such operations have to be done in a rather unnatural fashion using the GROUP BY operator.

Several syntactic normal forms have been proposed for nested relations [Embley et al. 1993; Mok et al. 1992; Ozsoyoglu and Yuan 1985; Ozsoyoglu and Yuan 1987a; Roth and Korth 1987] but, with the exception of the work by Embley *et al.*, most of the research has not addressed the issue of providing a formal justification for the normal forms and so this is an issue that needs further research. It is interesting to note that in respect to nested relations, normalising these relations can result in less fragmentation than for the flat relational model. For instance, if one takes the well known example [Date 1990] of a relation with attributes $COURSE$, $TEACHER$, $TEXT$ with the constraint that a *COURSE* can have multiple *TEACHERs* and multiple *TEXTs*, but the *TEACHERs* and *TEXTs* of a *COURSE* are independent, then this can be naturally modelled by allowing the attributes *TEACHER* and *TEXT* to be set valued. An instance of such a nested relation is shown below in Figure 7.4.

COURSE	TEACHER	TEXT
Physics	{Allan, Jones}	{Mechanics, Optics}
Maths	{Jones, Smith}	{Calculus, Algebra}

Figure 7.4. A nested relation

This relation has no processing difficulties and is normalised according to the definitions of nested normal forms in the works previously cited. However, when this nested relation is converted to a flat relation, the MVD $COURSE \rightarrow\rightarrow TEACHER$ holds in the flat relation and so violates 4NF. In order to avoid processing difficulties, the relation scheme must be split into *COURSE TEACHER* and *COURSE TEXT*.

Another generalisation of the relational model which has been the focus of considerable research are *deductive data models* [Ceri et al. 1989; Ullman 1985; Ullman

1988a; Ullman 1988b]. These models are based on *logic programming* [Lloyd 1987] and extend the power of relational systems by the incorporation of deductive processing. In most deductive data models, information is divided into two classes: *extensional (EDB) predicates* which correspond to tuples in relations and *intensional (IDB) predicates* which are rules defined in terms of EDB predicates or other IDB predicates.

Viewed in this framework, relations are simply sets of EDB predicates of the same type and the work in this thesis can be viewed as providing a framework for deciding how to structure the EDB predicates. However, some recent research [Debenham 1989; Debenham 1993] has demonstrated that some ways of structuring IDB predicates are inferior to others and problems may occur that are similar to the problems occurring in the EDB predicates. For instance, an IDB predicate may be implied by a set of other IDB predicates, or an IDB predicate may be able to be replaced by an equivalent IDB predicate with the same head but fewer elements in the body. So another area of research would be to investigate more formally what are the desirable structural properties of deductive database programs, as well as devising methods of detecting and converting incorrectly structured programs to correctly structured ones.

REFERENCES

Abiteboul, S. and Kanellakis, P. C. 1989. Object Identity as a Query Language Primitive. *Proceedings of the 1989 ACM SIGMOD International Conference on the Management of Data*, pp. 159-173.

Aho, A. V., Beeri, C. and Ullman, J. D. 1979a. The Theory of Joins in Relational Databases. *ACM Transactions on Database Systems, 4*, 3, 279-314.

Aho, A. V., Sagiv, Y. and Ullman, J. D. 1979b. Efficient Optimization of a Class of Relational Expressions. *ACM Transactions on Database Systems, 4*, 4, 435-454.

Aho, A. V., Sagiv, Y. and Ullman, J. D. 1979c. Equivalences among Relational Expressions. *SIAM Journal of Computing, 8*, 2, 218-246.

Armstrong, W. W. 1974. Dependency Structures of Data Base Relationships. *Proceedings of the IFIP Conference*, (Geneva), pp. 580-583.

Arora, A. K. and Carlson, C. R. 1978. The Information Preserving Properties of Certain Relational Database Transformations. *Proceedings of the 4th International Conference on Very Large Databases*, pp. 352-359.

Atzeni, P. and Chan, E. P. F. 1987. Independent Database Schemes under Functional and Inclusion Dependencies. *Proceedings of the 13th International Conference on Very Large Databases*, pp. 159-166.

Atzeni, P. and DeAntonellis, V. 1993. *Relational Database Theory.* Benjamin/Cummings, Redwood City, California.

Atzeni, P. and Morfuni, N. M. 1984. Functional Dependencies in Relations with Null Values. *Information Processing Letters, 18*, 4, 233-238.

Atzeni, P. and Parker, D. S. 1982. Assumptions in Relational Database Theory. *Proceedings of the 1st ACM SIGACT SIGMOD Symposium on Principles of Database Systems*, pp. 1-9.

Batini, C., Ceri, S. and Navathe, S. B. 1991. *Conceptual Database Design An Entity-Relationship Approach.* Benjamin/Cummings.

Bayer, R. and McCreight, E. M. 1972. Organization and Maintenance of Large Ordered Indices. *Acta Informatica, 1*, 3, 173-189.

Beeri, C. 1980. On the Membership Problem for Functional and Multivalued Dependencies in Relational Databases. *ACM Transactions on Database Systems, 5*, 3, 241-259.

Beeri, C. and Bernstein, P. A. 1979. Computational Problems Related to the Design of Normal Form Relational Schemas. *ACM Transactions on Database Systems, 4*, 1, 30-59.

Beeri, C., Bernstein, P. A. and Goodman, N. 1978. A Sophisticate's Introduction to Database Normalization Theory. *Proceedings of the 4th International Conference on Very Large Databases*, pp. 113-124.

Beeri, C. et al. 1983. On the Desirability of Acylic Database Schemes. *Journal of the Association for Computing Machinery, 30*, 3, 479-513.

Beeri, C., Fagin, R. and Howard, J. H. 1977. A Complete Axiomatization for Functional and Multivalued Dependencies in Database Relations. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 47-61.

Beeri, C. and Honeyman, P. 1981. Preserving Functional Dependencies. *SIAM Journal of Computing, 10*, 3, 647-656.

Beeri, C. and Kifer, M. 1986a. Elimination of Intersection Anomalies from Database Schemes. *Journal of the Association for Computing Machinery, 33*, 3, 423-450.

Beeri, C. and Kifer, M. 1986b. An Integrated Approach to Logical Design of Relational Database Schemes. *ACM Transactions on Database Systems, 11*, 2, 134-158.

Beeri, C. and Kifer, M. 1987. A Theory of Intersection Anomalies in Relational Database Schemes. *Journal of the Association for Computing Machinery, 34*, 3, 544-577.

Beeri, C. et al. 1981. Equivalence of Relational Database Schemes. *SIAM Journal of Computing, 10*, 2, 352-370.

Beeri, C. and Vardi, M. Y. 1981a. A Note on Decompositions of Relational Databases. *SIGMOD Record, 12*, 1, 33-37.

Beeri, C. and Vardi, M. Y. 1981b. On the Properties of Join Dependencies. In *Advances in Database Theory,* (Gallaire, H., Minker, J. and Nicolas, J. M., Ed.), Plenum Press, New York, pp. 25-72.

Beeri, C. and Vardi, M. Y. 1984a. Formal Systems for Tuple and Equality Generating Dependencies. *SIAM Journal of Computing, 13*, 1, 76-98.

Beeri, C. and Vardi, M. Y. 1984b. A Proof Procedure for Data Dependencies. *Journal of the Association for Computing Machinery, 31*, 4, 718-741.

Beeri, C. and Vardi, M. Y. 1985. Formal Systems for Join Dependencies. *Theoretical Computer Science, 38*, 99-116.

Bernstein, P. A. 1976. Synthesizing Third Normal Form Relations from Functional Dependencies. *ACM Transactions on Database Systems, 1*, 4, 277-298.

Bernstein, P. A. and Goodman, N. 1980. What Does Boyce-Codd Normal Form Do? *Proceedings of the 6th International Conference on Very Large Databases*, pp. 245-259.

Biller, H. 1979. On the Notion of Irreducible Relations. In *Database Architecture,* (Nijssen, G. M., Ed.), North-Holland, Amsterdam, pp. 277-295.

Biskup, J. 1989. Boyce-Codd Normal Form and Object Normal Form. *Information Processing Letters, 32*, 29-33.

Biskup, J., Dayal, U. and Bernstein, P. A. 1979. Synthesizing Independent Database Schemas. *Proceedings of the ACM SIGMOD International Conference on the Management of Data*, pp. 143-151.

Biskup, J. and Dublish, P. 1991. Objects in Relational Databases with Functional, Inclusion and Exclusion dependencies. *Proceedings of the 3rd Symposium on Mathematical Fundamentals of Database and Knowledge Base Systems*, (Rostock), pp. 276-290.

Casanova, M., Fagin, R. and Papadimitriou, C. H. 1984. Inclusion Dependencies and Their Interaction with Functional Dependencies. *Journal of Computer and System Sciences, 28*, 29-59.

Ceri, S., Gottolob, G. and Tanca, L. 1989. *Logic Programming*. Springer-Verlag, Berlin.

Chan, E. P. F. 1989. A Design Theory for Solving the Anomalies Problem. *SIAM Journal of Computing, 18*, 3, 429-448.

Chan, E. P. F. and Atzeni, P. 1992. On the Properties and Characterization of Connection-Trap-Free Schemes. *Journal of Computer and System Sciences, 44*, 1, 1-22.

Chan, E. P. F. and Hernàndez, H. J. 1988a. On Generating Database Schemes Bounded or Constant-time-maintainability by Extensibility. *Acta Informatica, 25*, 475-496.

Chan, E. P. F. and Hernàndez, H. J. 1988b. On the Desirability of γ-acyclic BCNF Database Schemes. *Theoretical Computer Science, 62*, 67-104.

Chan, E. P. F. and Hernàndez, H. J. 1991. Independence-Reducible Database Schemes. *Journal of the Association for Computing Machinery, 38*, 4, 854-886.

Chan, E. P. F. and Mendelzon, A. O. 1987. Independent and Separable Database Schemes. *SIAM Journal of Computing, 16*, 5, 841-851.

Chandra, A. K. and Vardi, M. Y. 1985. The Implication Problem for Functional and Inclusion Dependencies is Undecidable. *SIAM Journal of Computing, 14*, 3, 671-677.

Chen, P. P. 1976. The Entity-Relationship Model - Towards a Unified View of Data. *ACM Transactions on Database Systems, 1*, 1, 9-36.

Codd, E. F. 1970. A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM, 13*, 6, 377-387.

Codd, E. F. 1972. Further Normalization of the Database Relational Model. In *Courant Computer Science Symposia 6: Data Base Systems,* (Rustin, R., Ed.), Prentice-Hall, Englewood Cliffs, N.J., pp. 33-64.

Codd, E. F. 1974. Recent Investigations in Relational Database Systems. *Proceedings of the IFIP Conference*, (Stockholm, Sweden), pp. 1017-1021.

Codd, E. F. 1979. Extending the Database Relational Model to Capture More Meaning. *ACM Transactions on Database Systems, 4*, 4, 397-434.

Codd, E. F. 1986. Missing Information (Applicable and Inapplicable) in Relational Databases. *ACM SIGMOD Record, 15*, 4, 53-78.

Codd, E. F. 1987. More Commentary on Missing Information in Relational Databases (Applicable and Inapplicable). *ACM SIGMOD Record, 16*, 42-50.

Date, C. J. 1987. *A Guide to the SQL Standard.* Addison-Wesley.

Date, C. J. 1990. *Relational Database Writings 1985 -1989.* Addison-Wesley.

Date, C. J. 1990. *An Introduction to Database Systems.* Addison-Wesley.

Date, C. J. and Fagin, R. 1992. Simple Conditions for Guaranteeing Higher Normal Forms in Relational Databases. *ACM Transactions on Database Systems, 17*, 3, 465-476.

Dayal, U. and Bernstein, P. A. 1982. On the Correct Translation of Update Operations on Relational Views. *ACM Transactions on Database Systems, 7*, 3, 381-416.

Debenham, J. K. 1989. *Knowledge Systems Design.* Prentice-Hall.

Debenham, J. K. 1993. Normal Forms of Rule-based Knowledge Systems. *Knowledge-Based Systems, 2*, 3, 147-157.

DeBra, P. and Paradaens, J. 1982. Horizontal Decomposition and Their Impact on Query Solving. *SIGMOD Record, 13*, 1, 46-50.

DeBra, P. and Paredaens, J. 1983. An Algorithm for Horizontal Decompositions. *Information Processing Letters, 17*, 91-95.

DeBra, P. and Paredaens, J. 1990. Removing Redundancy and Updating Databases. *Proceedings of the 3rd International Conference on Database Theory*, (Paris), pp. 245-256.

Delobel, C. 1978. Normalization and Hierarchical Dependencies in the Relational Data Model. *ACM Transactions on Database Systems, 3*, 3, 201-222.

Delobel, C. and Adiba, M. 1985. *Relational Database Systems.* North Holland, Amsterdam.

Demetrovics, J. 1978. On the Number of Candidate Keys. *Information Processing Letters, 7*, 6, 226-269.

Demetrovics, J., Libkin, L. and Muchnik, I. B. 1992. Functional Dependencies in Relational Databases: A Lattice Point of View. *Discrete Applied Mathematics, 40*, 155-185.

Demetrovics, J. and Thi, V. D. 1988. Relations and Minimal Keys. *Acta Cybernetica, 8*, 3, 279-285.

Desai, B. C., Goyal, P. and Sadri, F. 1986. Updates in Relational Databases. *NCC 86, AFIPS Proceedings, 55*, 237-244.

Desai, B. C., Goyal, P. and Sadri, F. 1987. Fact Structure and its Application to Updates in Relational Databases. *Information Systems, 12*, 2, 215-221.

Diederich, J. and Milton, J. 1988. New Methods and Fast Algorithms for Database Normalization. *ACM Transactions on Database Systems, 13*, 3, 339-365.

El-Masri, R. A. and Navathe, S. B. 1989. *Fundamentals of Database Systems.* Benjamin/Cummings.

Embley, D., Ng, Y. and Mok, W. Y. 1993. Unifying Normalization Theory Under a New Definition for Nested Normal Form. Report No. BYU-CS-93-1, Department of Computer Science, Bringham Young University.

Fagin, R. 1977a. The Decomposition Versus the Synthetic Approach to Relational Database Design. *Proceedings of the 3rd International Conference on Very Large Databases*, pp. 441-446.

Fagin, R. 1977b. Functional Dependencies in a Relational Database and Propositional Logic. *IBM Journal of Research and Development, 21*, 6, 534-544.

Fagin, R. 1977c. Multivalued Dependencies and a New Normal Form for Relational Databases. *ACM Transactions on Database Systems, 2*, 3, 262-278.

Fagin, R. 1979. Normal Forms and Relational Database Operators. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 153-160.

Fagin, R. 1981. A Normal Form for Relational Databases that is based on Domains and Keys. *ACM Transactions on Database Systems, 6*, 3, 387-415.

Fagin, R. 1983. Degrees of Acylicity for Hypergraphs and Relational Database Schemes. *Journal of the Association for Computing Machinery, 30*, 3, 514-550.

Fagin, R. et al. 1986. Updating Logical Databases. In *Advances in Computing Research,* (Kanellakis, P. C. and Preparata, F. P., Ed.), JAI Press, London, pp. 1-18.

Fagin, R., Mendelzon, A. O. and Ullman, J. D. 1982. A Simplified Universal Relation Assumption and its Properties. *ACM Transactions on Database Systems, 7*, 3, 343-360.

Fagin, R. et al. 1979. Extendible Hashing - A Fast Access Method for Dynamic Files. *ACM Transactions on Database Systems, 4*, 3, 315-344.

Fagin, R., Ullman, J. D. and Vardi, M. Y. 1983. On the Semantics of Updates in Databases. *Proceedings of the 2nd ACM SIGACT SIGMOD Symposium on the Principles of Database Systems*, pp. 352-365.

Forsyth, J. and Fadous, R. 1975. Finding Candidate Keys for Relational Databases. *Proceedings of the ACM SIGMOD International Conference on the Management of Data*, pp. 203-210.

Furtado, A. L. 1981. Horizontal Decompositions to Improve a non-BCNF Scheme. *SIGMOD Record, 12*, 1, 26-32.

Furtado, A. L. and Casanova, M. A. 1985. Updating Relational Views. In *Query Processing in Database Systems,* (Kim, W., Reiner, D. S. and Batory, D. S., Ed.), Springer-Verlag.

Galil, Z. 1982. An Almost Linear Time Algorithm for Computing the Dependency Basis in a Relational Database. *Journal of the Association for Computing Machinery, 29*, 1, 96-102.

Garey, M. R. and Johnson, D. S. 1979. *Computers and Intractibility.* San Francisco, W. H. Freeman and Company.

Ginsburg, S. and Zaiddan, S. M. 1982. Properties of Functional Dependency Families. *Journal of the Association for Computing Machinery, 29*, 3, 678-698.

Graham, M. H., Mendelzon, A. O. and Vardi, M. Y. 1986. Notions of Dependency Satisfaction. *Journal of the Association for Computing Machinery, 33*, 1, 105-129.

Graham, M. H. and Yannakis, M. 1984. Independent Database Schemas. *Journal of Computer and System Sciences, 28*, 121-141.

Grahne, G. and Raiha, K. J. 1983. Database Decomposition into Fourth Normal Form. *Proceedings of the 9th International Conference on Very Large Databases*, pp. 186-196.

Gurevich, Y. 1982. The Inference Problem for Template Dependencies. *Information and Control, 55*, 69-79.

Hagihara, K. et al. 1979. Decision Problems for Multivalued Dependencies in Relational Databases. *SIAM Journal of Computing, 8*, 2, 247-264.

Hall, P., Owlett, T. and Todd, S. 1976. Relations and Entities. In *Modelling in Database Management Systems,* (Nijssen, G. M., Ed.), North-Holland, Amsterdam.

Heath, I. J. 1971. Unacceptable File Operations in a Relational Database. *Proceedings of the ACM SIGFIDET Workshop on Data Description, Access and Control*, pp. 19-33.

Hernàndez, H. J. and Chan, E. P. F. 1991. Constant-Time-Maintainable BCNF Database Schemes. *ACM Transactions on Database Systems, 16*, 4, 571-599.

Honeyman, P. 1980. Functional Dependencies and the Universal Instance Property in the Relational Model of Database Systems. Ph.D. Thesis, Princeton University, Princeton NJ.

Honeyman, P. 1982. Testing Satisfaction of Functional Dependencies. *Journal of the Association for Computing Machinery, 29*, 3, 668-677.

Honeyman, P., Ladner, R. E. and Yannakis, M. 1980. Testing the Universal Instance Assumption. *Information Processing Letters, 10*, 1, 14-19.

Hull, R. B. 1986. Relative Information Capacity of Simple Relational Database Schemata. *SIAM Journal of Computing, 15*, 3, 856-886.

Ito, M. et al. 1984. Membership Problems for Data Dependencies in Relational Expressions. *Theoretical Computer Science, 34*, 315-335.

Ito, M., Taniguchi, K. and Kasami, T. 1983. Membership Problem for Embedded Multivalued Dependencies under some Restricted Conditions. *Theoretical Computer Science, 22*, 175-194.

Jaeschke, G. and Schek, H. J. 1982. Remarks on the Algebra for Non First Normal Relations. *Proceedings of the 1st ACM SIGACT SIGMOD Symposium on Principles of Database Systems*, pp. 124-138.

Jajodia, S. 1986. Recognizing Multivalued Dependencies in Relation Schemas. *The Computer Journal, 29*, 5, 458-459.

Jajodia, S. and Ng, P. A. 1983. Update Sets Approach to Databases. *Proceedings of the IEEE 7th International Computer Software and Applications Conference*, pp. 194-200.

Jou, J. H. and Fischer, P. C. 1983. The Complexity of Recognizing 3NF Schemes. *Information Processing Letters, 14*, 4, 187-190.

Kambayashi, Y. 1979. Equivalent key problem of the relational database model. In *Lecture Notes in Computer science No. 85,* Springer-Verlag, Berlin, pp. 165-192.

Kambayashi, Y., Tanaka, K. and Yajima, S. 1979. Semantic Aspects of Data Dependencies and their Application to Relational Database Design. *Proceedings of the COMPSAC 79*, pp. 398-403.

Kandzia, P. and Manglemann, M. 1980. On Covering Boyce-Codd Normal Forms. *Information Processing Letters, 11*, 4,5, 218-223.

Kanellakis, P. C. 1980. On the Computational Complexity of Cardinality Constraints in Relational Databases. *Information Processing Letters, 11*, 2, 98-101.

Katsuno, H. 1981. On Two Different Meanings of Multivalued Dependencies in a Conceptual Schema. *The Transactions of the IECE of Japan, 64*, 6, 383-389.

Kent, W. 1981. Consequences of Assuming a Universal Relation. *ACM Transactions on Database Systems, 6*, 4, 539-556.

Khosafian, S. N. and Copeland, G. P. 1986. Object Identity. *Proceedings of the OOPSLA Conference*, pp. 406-417.

Kim, W. 1990. Object-Oriented Databases : Definition and Research Directions. *IEEE Transactions on Knowledge and Data Engineering, 2*, 3, 327-341.

Korth, H. F. et al. 1984. System/U: A Database System Based on the Universal Relation Assumption. *ACM Transactions on Database Systems, 9*, 3, 331-347.

Lakshmanan, V. S. and VeniMadhavan, C. E. 1985. The Implication Problem for Functional and Multivalued Dependencies : An Algebraic Approach. *Proceedings of the 5th International Conference on Foundations of Software Technology and Theoretical Computer Science*, (New Delhi), pp. 303-328.

Lakshmanan, V. S. and VeniMadhavan, C. E. 1987. An Algebraic Theory of Functional and Multivalued Dependencies in Relational Databases. *Theoretical Computer Science, 54*, 103-128.

LeDoux, C. H. and Parker, D. S. 1982. Reflections on Boyce-Codd Normal Form. *Proceedings of the 8th International Conference on Very Large Databases*, pp. 131-141.

Lee, T. T. 1983. An Algebraic Theory of Relational Databases. *The Bell System Technical Journal, 62*, 10, 3159 - 3204.

Levene, M. and Loizou, G. 1989. NURQL: A Nested Universal Query Language. *Information Systems, 14*, 307-316.

Lien, Y. E. 1979. Multivalued Dependencies with Null values in Relational Databases. *Proceedings of the 5th International Conference on Very Large Databases*, pp. 61-66.

Lien, Y. E. 1981. Hierarchical Scemata for Relational Databases. *ACM Transactions on Database Systems, 6*, 1, 48-69.

Lien, Y. E. 1982. On the Equivalence of Database Models. *Journal of the Association for Computing Machinery, 29*, 2, 333-362.

Ling, T. 1985. An Analysis of Multivalued and Join Dependencies based on the Entity-Relationship Approach. *Data and Knowledge Engineering, 1*, 253-271.

Ling, T., Tompa, F. W. and Kameda, T. 1981. An Improved Third Normal Form for Relational Databases. *ACM Transactions on Database Systems, 6*, 2, 329-346.

Lloyd, J. W. 1987. *Foundations of Logic Programming.* Springer-Verlag, Berlin.

Lucchesi, C. L. and Osborn, S. L. 1978. Candidate Keys for Relations. *Journal of Computer and System Sciences, 17*, 2, 270-279.

Maier, D. 1983. *The Theory of Relational Databases.* Computer Science Press, Rockville, Md.

Maier, D. et al. 1980. Adequacy of Decompositions in Relational Databases. *Journal of Computer and System Sciences, 21*, 3, 368-379.

Maier, D., Mendelzon, A. O. and Sagiv, Y. 1979. Testing Implications of Data Dependencies. *ACM Transactions on Database Systems, 4*, 4, 455-469.

Maier, D. et al. 1987. PIQUE: A Relational Query Language without Relations. *Information Systems, 12*, 3, 317-355.

Maier, D., Rozenshtein, D. and Warren, D. S. 1983. Windows on the World. *Proceedings of the ACM SIGMOD International Conference on the Management of Data*, pp. 68-78.

Maier, D., Rozenshtein, D. and Warren, D. S. 1985. Representing Roles in Universal Scheme Interfaces. *IEEE Transactions on Software Engineering, SE-11*, 7, 644-652.

Maier, D., Rozenshtein, D. and Warren, D. S. 1986. Window Functions. In *Advances in Computing Research,* (Kanellakis, P. C. and Preparata, F. P., Ed.), JAI Press, Greenwich, pp. 213-246.

Maier, D., Sagiv, Y. and Yannakis, M. 1981. On the Complexity of Testing Implications of Functional and Join Dependencies. *Journal of the Association for Computing Machinery, 28*, 4, 680-695.

Maier, D. and Ullman, J. D. 1983. Maximal Objects and the Semantics of Universal Relation Databases. *ACM Transactions on Database Systems, 8*, 1, 1-14.

Maier, D., Ullman, J. D. and Vardi, M. Y. 1984. On the Foundations of the Universal Relation Model. *ACM Transactions on Database Systems, 9*, 2, 283-308.

Maier, D. and Warren, D. S. 1982. Specifying Connections for a Universal Relation Scheme Database. *Proceedings of the ACM SIGMOD International Conference on the Management of Data*, pp. 1-7.

Makinouchi, A. 1977. A Consideration on Normal Form of not necessarily Normalized Relations in the Relational Data Model. *Proceedings of the 3rd International Conference on Very Large Databases*, pp. 447-453.

Matus, F. 1991. Abstract Functional Dependency Structures. *Theoretical Computer Science, 81*, 1, 117-126.

Mendelzon, A. O. 1984. Database States and their Tableaux. *ACM Transactions on Database Systems, 9*, 2, 264-282.

Mitchell, J. C. 1983a. The Implication Problem for Functional and Inclusion Dependencies. *Information and Control, 56*, 154-173.

Mitchell, J. C. 1983b. Inference Rules for Functional and Inclusion Dependencies. *Proceedings of the 2nd ACM SIGACT SIGMOD Symposium on Principles of Database Systems*, pp. 58-69.

Mok, W. K., Ng, Y. K. and Embley, D. W. 1992. An Improved Nested Normal Form for Use in Object-Oriented Software Systems. *Proceedings of the 2nd International Computer Science Conference - Data and Knowledge Engineering: Theory and Applications*, (Hong Kong), pp. 446-452.

Nakamura, F. and Chen, P. P. 1981. Semantic Considerations on Multivalued Dependencies in Relational Databases. *Journal of Information Processing, 4*, 3, 134-141.

Nijssen, G. M. 1977. On the Gross Architecture for the Next Generation Database Management Systems. In *Information Processing '77,* (Gilchrist, B., Ed.), North-Holland, Amsterdam.

Nijssen, G. M. 1979. *Architecture and Models in Database Management Systems.* North-Holland, Amsterdam.

Nijssen, G. M. and Halpin, T. A. 1989. *Conceptual Schema and Relational Database Design: A Fact-Based Approach.* Prentice-Hall.

Novotny, J. and Novotny, M. 1992. Notes on the Algebraic Approach to Dependence in Information Systems. *Fundamenta Informaticae, 16*, 3-4, 263-273.

Ozsoyoglu, Z. M. and Yuan, L. Y. 1985. A Normal Form for Nested Relations. *Proceedings of the 4th ACM SIGACT SIGMOD Symposium on Principles of Database Systems*, pp. 251-260.

Ozsoyoglu, Z. M. and Yuan, L. Y. 1987a. A New Normal Form for Nested Relations. *ACM Transactions on Database Systems, 12*, 1, 111-136.

Ozsoyoglu, Z. M. and Yuan, L. Y. 1987b. Reduced MVDs and Minimal Covers. *ACM Transactions on Database Systems, 12*, 3, 377-394.

Paredaens, J. et al. 1989. *The Structure of the Relational Database Model.* Springer-Verlag, Berlin.

Parker, D. S. and Delobel, C. 1979. Algorithmic Applications for a New Result on Multivalued Dependencies. *Proceedings of the 5th International Conference on Very Large Databases*, pp. 67-74.

Parker, D. S. and Parsaye-Ghomi, K. 1980. Inferences Involving Embedded Multivalued Dependencies and Transitive Dependencies. *Proceedings of the ACM SIGMOD International Conference on the Management of Data*, pp. 52-57.

Petrov, S. V. 1989. Finite Axiomatization of Languages for Representation of System Properties: Axiomatization of Dependencies. *Information Sciences, 47*, 339-372.

Pichat, E. 1985. Algorithms For the Decomposition of Keys. *R. A. I. R. O. Theoretical Informatics, 19*, 3, 213-232.

Rissanen, J. 1977. Independent Components of Relations. *ACM Transactions on Database Systems, 2*, 4, 317-315.

Rissanen, J. 1979. Theory of Joins for Relational Databases - A Tutorial Survey. In *Mathematical Foundations of Computer Science, Lecture Notes in Computer Science 64*), Springer-Verlag, Berlin, pp. 537-551.

Rissanen, J. 1982. On Equivalence of Database Schemes. *Proceedings of the 1st ACM SIGACT SIGMOD Symposium on Principles of Database Systems*, pp. 23-26.

Roth, M. A. and Korth, H. F. 1987. The Design of ¬1NF Relational databases into Nested Normal Form. *Proceedings of the ACM SIGMOD International Conference on the Management of Data*, pp. 143-159.

Roth, M. A., Korth, H. F. and Batory, D. S. 1987. SQL/NF: A Query Language for ¬1NF Relational Databases. *Information Systems, 12*, 1, 99-114.

Roth, M. A., Korth, H. F. and Silberschatz, A. 1988. Extended Albebra and Calculus for ¬1NF Relational Databases. *ACM Transactions on Database Systems, 13*, 4, 389-417.

Sagiv, Y. 1980. An Algorithm for Inferring Multivalued Dependencies with an Application to Propositional Logic. *Journal of the Association for Computing Machinery, 27*, 2, 250-262.

Sagiv, Y. 1981. Can We Use the Universal Relation Instance Assumption Without Using Nulls? *Proceedings of the ACM SIGMOD International Conference on the Management of Data*, pp. 108-120.

Sagiv, Y. 1983. A Characterization of Globally Consistent Databases and their Correct Access Paths. *ACM Transactions on Database Systems, 8*, 2, 266-286.

Sagiv, Y. 1988. On Bounded Database Schemes and Bounded Horn-Clause Programs. *SIAM Journal of Computing, 17*, 1, 1-22.

Sagiv, Y. et al. 1981. An Equivalence Between Relational Databases and a Fragment of Propositional Logic. *Journal of the Association for Computing Machinery, 28*, 3, 435-453.

Sagiv, Y. and Walecka, S. 1982. Subset dependencies and a Completeness Result for a Subclass of Embedded Multivalued Dependencies. *Journal of the Association for Computing Machinery, 29*, 1, 103-17.

Schek, H. J. and Scholl, M. H. 1986. The Relational Model with Relation-Valued Attributes. *Information Systems, 11*, 2, 137-147.

Sciore, E. 1980. The Universal Interface and Database Design. Ph.D. Thesis, Princeton University, Princeton NJ.

Sciore, E. 1981. Real-World MVDs. *Proceedings of the ACM SIGMOD International Conference on the Management of Data*, pp. 121-132.

Sciore, E. 1982. A Complete Axiomatization of Full Join Dependencies. *Journal of the Association for Computing Machinery, 29*, 2, 373-393.

Sciore, E. 1983a. Improving Database Schemes by Adding Attributes. *Proceedings of the 2nd ACM SIGACT SIGMOD Symposium on Principles of Database Systems*, pp. 379-384.

Sciore, E. 1983b. Inclusion Dependencies and the Universal Instance. *Proceedings of the 2nd ACM SIGACT SIGMOD Symposium on Principles of Database Systems*, pp. 48-57.

Selesnjew, O. and Thalheim, B. 1988. On the Number of Shortest Keys in Relational Databases on Nonuniform Domains. *Acta Cybernetica, 8*, 3, 267-271.

Smith, J. M. 1978. A Normal Form for Abstract Syntax. *Proceedings of the 4th International Conference on Very Large Databases*, pp. 156-162.

Tanaka, K., Kambayashi, Y. and Yajima, S. 1979. Properties of Embedded Multivalued Dependencies in Relational Databases. *The Transactions of the IECE of Japan, 62*, 8, 536-543.

Thalheim, B. 1988. Open problems in Database Theory. In *Proceedings 1st Symposium on Mathematical Fundamentals of Database Systems, Lecture Notes in Computer Science no. 305*), Springer Verlag, pp. 241-247.

Thalheim, B. 1991. *Dependencies in Relational Databases.* B. G. Teubner.

Thalheim, B. 1992. The Number of Keys in Relational and Nested Relational Databases. *Discrete Applied Mathematics, 40*, 265-282.

Thalheim, B. and Al-Fedhagi, S. 1990. Preserving two-tuple Dependencies under Projection. *Acta Cybernetica, 9*, 4, 441-458.

Thuan, H. 1987. Some Remarks on the Algorithm of Lucchesi and Osborn. *Acta Cybernetica, 8*, 2, 191-193.

Thuan, H. and Bao, L. V. 1985. Some Results about Keys of Relational Schemes. *Acta Cybernetica, 7*, 1, 99-113.

Ullman, J. D. 1982. The U.R. Strikes Back. *Proceedings of the 1st ACM SIGACT SIGMOD Symposium on Principles of Database Systems*, pp. 10-22.

Ullman, J. D. 1983a. On Kent's 'Consequences of Assuming a Universal Relation'. *ACM Transactions on Database Systems, 8*, 4, 637-643.

Ullman, J. D. 1983b. Universal Relation Interfaces for Database Systems. In *Information Processing 83,* (Mason, R. E. A., Ed.), North-Holland, pp. 243-252.

Ullman, J. D. 1985. Implementation of a Logical Query Languages for Databases. *ACM Transactions on Database Systems, 10*, 3, 289-321.

Ullman, J. D. 1987. Database Theory - Past and Future. *Proceedings of the 6th ACM SIGACT SIGMOD Conference on Principles of Database Systems*, pp. 1-10.

Ullman, J. D. 1988a. *Principles of Database and Knowledge-Base Systems.* Vol. 1. Computer Science Press.

Ullman, J. D. 1988b. *Principles of Database and Knowledge-Base Systems.* Vol. 2. Computer Science Press.

Vardi, M. Y. 1988. Fundamentals of Dependency Theory. In *Trends in Theoretical Computer Science,* (Borger, E., Ed.), Computer Science Press, Rockville, pp. 171-224.

Vardi, M. Y. 1983. Inferring Multivalued Dependencies From Functional and Join Dependencies. *Acta Informatica, 19*, 2, 305-324.

Vardi, M. Y. 1984. The Implication and Finite Implication Problems For Typed Template Dependencies. *Journal of Computer and System Sciences, 28*, 1, 3-28.

Vassilou, Y. 1979. Null Values in Database Management - A Denotational Semantics Approach. *Proceedings of the ACM SIGMOD International Conference on the Management of Data*, pp. 162-169.

Vassilou, Y. 1980. Functional Dependencies and Incomplete Information. *Proceedings of the 6th International Conference on Very Large Databases*, pp. 260-269.

Vincent, M. W. 1991. Equivalence of Update Anomalies in Relational Databases. In *Advances in Data Management - Proceedings COMAD 3rd International Conference on Data Management,* (Sadanandan, P. and Vijyaraman, T. M., Ed.), Tata McGraw-Hill, New Delhi, pp. 181 - 197.

Vincent, M. W. 1992a. An Efficient Method for Testing 4NF in Relational Databases. *Australian Computer Science Communications, 14*, 1, 955-966.

Vincent, M. W. 1992b. Modification Anomalies and Boyce-Codd Normal Form. In *Research and Practical Issues in Data Management,* (Srinivasan, B. and Zeleznikow, J., Ed.), World Scientific Press, pp. 251-264.

Vincent, M. W. and Srinivasan, B. 1992a. Redundancy and the Justification for Fourth Normal Form in Relational Databases. *Proceedings of the 2nd International Computer*

Science Conference - Data and Knowledge Engineering: Theory and Applications, (Hong Kong), pp. 432-438.

Vincent, M. W. and Srinivasan, B. 1992b. Semantic Justification for 4NF in Relational Database Design. In *Computer and Data Management : Proceedings COMAD 4th International Conference on the Management of Data,* (Kaujalgi, V. B. and Krishnamurthy, H., Ed.), Tata McGraw-Hill, New Delhi, pp. 51-60.

Vincent, M. W. and Srinivasan, B. 1993a. Armstrong Databases for Functional and Multivalued Dependencies in Relational Databases. In *Advances in Database Research,* (Orlowska, M. E. and Papazologu, M., Ed.), World Scientific Press, Singapore, pp. 317-328.

Vincent, M. W. and Srinivasan, B. 1993b. Fact-Based Update Anomalies and Normal Forms in Relational Database Design. *Australian Computer Science Communications, 15*, 1, 665-673.

Vincent, M. W. and Srinivasan, B. 1993c. The Preservation of Key Values and a New Normal Form for Relational Databases. Submitted for publication.

Vincent, M. W. and Srinivasan, B. 1993d. Update Anomalies and 4NF in Relational Databases - the MVD case. Report No. 93/181, Monash University.

Vincent, M. W. and Srinivasan, B. 1994a. Key-Based Update Anomalies and the Justification for 4NF in Database Design. In *ADC'94,* (Sacks-Davis, R. Ed.), Global Publication Services, pp. 346-359.

Vincent, M. W. and Srinivasan, B. 1994b. A Note on Relation Schemes which are in 3NF but not in BCNF. To appear in *Information Processing Letters.*

Vincent, M. W. and Srinivasan, B. 1994c. Redundancy and the Justification for Fourth Normal Form in Relational Databases. To appear in *International Journal of Foundations of Computer Science.*

Vincent, M. W. and Srinivasan, B. 1994d. Update Anomalies and the Justification for 4NF in Relational Databases. To appear in *Information Sciences.*

Vossen, G. 1988. A New Characterization of Fd Implication with an Application to Update Anomalies. *Information Processing Letters, 29*, 131-135.

Vossen, G. 1990. *Data Models, Database Languages and Database Management Systems.* Addison-Wesley.

Yang, C. 1986. *Relational Databases.* Englewood Cliffs, N. J., Prentice-Hall.

Yu, C. T. and Johnson, D. T. 1976. On the Complexity of Finding the Set of Candidate Keys for a Given Set of Functional Dependencies. *Information Processing Letters, 5*, 4, 100-101.

Yuan, L. Y. and Ozsoyoglu, Z. M. 1986. Unifying Functional and Multivalued Dependencies for Relational Database Design. *Proceedings of the 5th ACM SIGACT SIGMOD Symposium on Principles of Database Systems*, pp. 183-190.

Yuan, L. Y. and Ozsoyoglu, Z. M. 1987. Logical design of Relational Database Schemes. *Proceedings of the 6th ACM SIGACT SIGMOD Symposium on Principles of Database Systems*, pp. 38-47.

Yuan, L. Y. and Ozsoyoglu, Z. M. 1992a. Design of Desirable Database Schemes. *Journal of Computer and System Sciences, 45*, 3, 435-470.

Yuan, L. Y. and Ozsoyoglu, Z. M. 1992b. Unifying Functional and Multivalued Dependencies for Relational Database Design. *Information Sciences, 59*, 189-211.

Zaniolo, C. 1976. Analysis and Design of Relational Schemata for Database Systems. Ph.D. Thesis, UCLA, Los Angeles, California.

Zaniolo, C. 1982. A New Normal Form for the Design of Relational Database Schemata. *ACM Transactions on Database Systems, 7*, 3, 489-499.

Zaniolo, C. 1984. Database Relations with Null Values. *Journal of Computer and System Sciences, 28*, 1, 142-166.

Zaniolo, C. and Melankoff, M. A. 1981. On the Design of Relational Database Schemata. *ACM Transactions on Database Systems, 6*, 1, 1-47.

Zaniolo, C. and Melankoff, M. A. 1982. A Formal Approach to the Definition and the Design of Conceptual Schemata for Database Systems. *ACM Transactions on Database Systems, 7*, 1, 1-23.