

Efficiency measurements between the third and fourth normal forms of database schemes
by Hirohisa Shimura

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science
Montana State University
© Copyright by Hirohisa Shimura (1987)

Abstract:
The purpose of this thesis is to explore the relationship between the efficiency of the third and the fourth normal forms in a database scheme and to prove the fourth normal form is more efficient and easier to use than the third normal form if the unnormalized form scheme contains a multivalued dependency.

Three methodologies are used to measure the efficiency of the third and the fourth normal form schemes. The first methodology measures the efficiency from the mathematical standpoint. The second methodology measures the efficiency from the machine standpoint, which is the machine time and the memory space required to execute a query. The third methodology measures the efficiency from the end user standpoint which is measured by counting the number of relational operations and execution time needed to execute a query.

First, if multivalued dependencies are present but not detected within the data model, a substantial amount of data redundancy will be expected to occur. If the third normal form scheme is utilized, the mathematical proof has shown that the memory space increases in size in a multiplicative fashion as the complexity of the data relationship increases. If the fourth normal form scheme is utilized, the memory space will increase in an additive fashion as the complexity of the data entities increases.

Second, during the machine efficiency measurements, the fourth normal form scheme performed more efficiently than the third normal form scheme in terms of the memory space utilization and the execution time. These observations were made during the machine simulation that encompassed four query exercises.

Third, observations regarding the relative ease of use associated with a series of end user queries has shown that the fourth normal form data scheme tended to be an easier database scheme to use. Ease of use was measured by the number of relational operations employed and execution time associated with a completed query routine.

Furthermore, the user's cognitive style has been shown to be an important factor in measuring the relative ease of use of a query.

The user's cognitive style may become a variable of interest during the database design phase. This should allow the database system to perform more efficiently when designed to incorporate the user's cognitive traits.

The thesis proves that the fourth normal form scheme is relatively more efficient and easier to use than the third normal form in terms of 1) the memory space allocation and utilization of both the theoretical and the machine stand points, 2) the machine execution time, and 3) the end user efficiency measurement measured by the number of relational operations employed and execution time associated with a completed query routine.

# EFFICIENCY MEASUREMENTS BETWEEN THE THIRD AND FOURTH NORMAL FORMS OF DATABASE SCHEMES

by

Hirohisa Shimura

A thesis submitted in partial fulfillment
of the requirements for the degree

of

Master of Science

in

Computer Science

MONTANA STATE UNIVERSITY
Bozeman, Montana

November 1987

APPROVAL

of a thesis submitted by

Hirohisa Shimura

This thesis has been read by each member of the thesis committee and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the College of Graduate Studies.

November 13th 1987
Date

J Dunbigh Stanley
Chairperson, Graduate Committee

Approved for the Major Department

November 13th 1987
Date

J Dunbigh Stanley
Head, Major Department

Approved for the College of Graduate Studies

December 18, 1987
Date

Harry L Parsons
Graduate Dean

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a master's degree at Montana State University, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his absence, by the Dean of Libraries when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature Heather Stimson

Date 11-13-87

TABLE OF CONTENTS

1. INTRODUCTION..... 1

2. DATABASE MANAGEMENT SYSTEMS..... 5
Introduction..... 5
Fundamental Database Models..... 7
Hierarchical Database Model..... 7
Network Database Model..... 8
Relational Database Model..... 9

3. NORMALIZATION THEORY AND NORMAL FORMS..... 14
Introduction..... 14
Normalization Theory..... 14
Four Criteria of Normalization..... 16
Dependencies and Redundancies..... 17
Functional Dependency..... 19
Multivalued Dependency..... 20
Join Dependency..... 21
Normal Forms..... 23

4. DATA INTEGRITY OF THE THIRD AND FOURTH NORMAL FORMS..... 32
Introduction..... 32
Data Integrity..... 32
Maintenance Policies..... 33
Disjoint Format Policy..... 35
Random Mix Policy..... 35
Cross Product Policy..... 36
Data Independence..... 37
The Third and Fourth Normal Forms..... 39

5. MATHEMATICAL VIEW OF STORAGE SPACE EFFICIENCY OF THE THIRD AND FOURTH NORMAL FORMS..... 46
Introduction..... 46
The Third Normal Form as a Multiplicative Model..... 47
Basis Step..... 47
Induction Hypothesis..... 48
Induction Step..... 49
The Fourth Normal Form as an Additive Model..... 50
Basis Step..... 50
Induction Hypothesis..... 51
Induction Step..... 52

6. MACHINE TIME EFFICIENCY MEASUREMENTS..... 54
Introduction..... 54
Need for the Machine Efficiency Measurement..... 54
Space Efficiency Measurement Methodologies..... 57
Analysis of Space Efficiency Measurements..... 60
Analysis of the First Query Routine..... 61
Analysis of the Second Query Routine..... 63
Analysis of the Third Query Routine..... 64
Analysis of the Fourth Query Routine..... 65
Methodology for Time Efficiency Measurements..... 66
Analysis of Time Efficiency Measurements..... 70
Analysis of the First Query Routine..... 70
Analysis of the Second Query Routine..... 71
Analysis of the Third Query Routine..... 71
Analysis of the Fourth Query Routine..... 72

7. EFFICIENCY MEASUREMENTS OF USERS OF THIRD AND FOURTH NORMAL FORMS..... 73
Introduction..... 73
Methodology for User's Response Time Measurements..... 74
Data Description..... 75
Data Analysis..... 79
Cognitive Style Methodology..... 80
Analysis of the Four Query Routines..... 82
Analysis of the First Query Routine..... 82
Analysis of the Second Query Routine..... 86
Analysis of the Third Query Routine..... 90
Analysis of the Fourth Query Routine..... 94

8. CONCLUSION.....104

BIBLIOGRAPHY.....106

APPENDICES.....111
APPENDIX A. THE QUERY ACCESS PATHS FOR THE FOUR QUERY ROUTINES.....112
APPENDIX B. MEMORY ALLOCATIONS FOR THE FOUR QUERY ROUTINES.....119
APPENDIX C. THE EXECUTION TIMES FOR THE FOUR QUERY ROUTINES.....124
APPENDIX D. QUESTIONNAIRE.....129
APPENDIX E. DATA.....134

# LIST OF TABLES


| Table | | Page |
|---|---|---|
| 1. | The Disjoint Format Maintenance Policy | 35 |
| 2. | The Random Mix Maintenance Policy | 36 |
| 3. | The Cross Product Maintenance Policy | 37 |
| 4. | A Data Dependent Record | 38 |
| 5. | Functional and Multivalued Dependencies | 44 |
| 6. | Basis Step of the Third Normal Form Proof | 47 |
| 7. | Induction Hypothesis of the Third Normal Form Proof | 48 |
| 8. | Induction Step of the Third Normal Form Proof | 50 |
| 9. | Basis Step of the Fourth Normal Form Proof | 51 |
| 10. | Induction Hypothesis of the Fourth Normal Form Proof | 52 |
| 11. | Induction Step of the Fourth Normal Form Proof | 53 |
| 12. | The Complexity of Data Relationship | 58 |
| 13. | Machine Time Comparison of Two Approaches to One Query Measured in units of 1/100 second | 69 |
| 14. | The Distribution of the User Group | 75 |
| 15. | The Distribution of User's Sex | 75 |
| 16. | The Distribution of User's School Year | 76 |
| 17. | The Distribution of the Normalized Form Schemes | 76 |
| 18. | A List of Variables for Descriptive Analysis | 76 |
| 19. | Academic Background Versus Normalized Forms | 77 |
| 20. | Computer Background Versus Normalized Forms | 78 |

21. Normalized Forms Versus Users..... 78

22. Users Versus Academic Background..... 78

23. Users Versus Computer Background..... 78

24. A List of Variables Used to Analyze User Response Time and the Number of Operations..... 79

25. The Initial Analysis of the First Query Routine Based on the Number of Relational Operations..... 83

26. The Detailed Analysis of the First Query Routine Based on the Number of Relational Operations..... 83

27. Machine Time Comparison Between the Third And the Fourth Normal Form Schemes During the First Query Routine..... 84

28. Time Comparison Between the Normalized Schemes During the First Query Routine..... 84

29. The Initial Analysis of the First Query Routine Based on the Execution Time..... 84

30. The Detailed Analysis of the First Query Routine Based on the Execution Time..... 85

31. Time Comparison Between Academic Majors and Cognitive Style 2..... 86

32. The Initial Analysis of the Second Query Routine Based on the Number of Relational Operations..... 87

33. The Detailed Analysis of the Second Query Routine Based on the Number of Relational Operations..... 87

34. The Comparison of the Number of Relational Operations Between Academic Majors During the Second Query Routine..... 87

35. Machine Time Comparison Between the Third and the Fourth Normal Form Schemes During the Second Query Routine..... 87

36. The Initial Analysis of the Second Query Routine Based on the Execution Time..... 88

37. The Detailed Analysis of the Second Query Routine Based on the Execution Time..... 88

38. Time Comparison Between the Normalized Form Schemes During the Second Query Routine..... 89

39. Time Comparison Between Academic Majors..... 89

40. The Comparison of the Number of Relational Operations Between the Normalized Form Schemes During the Third Query Routine..... 90

41. The Initial Analysis of the Third Query Routine Based on the Number of Relational Operations..... 91

42. The Detailed Analysis of the Third Query Routine Based on the Number of Relational Operations..... 91

43. The Comparison of the Number of Relational Operations Between Academic Majors and the Normalized Forms..... 91

44. Machine Time Comparison Between the Third And the Fourth Normal Form Schemes During the First Query Routine..... 92

45. The Initial Analysis of the Third Query Routine Based on the Execution Time..... 93

46. The Detailed Analysis of the Third Query Routine Based on the Execution Time..... 93

47. Time Comparison Between the Normalized Form Schemes During the Third Query Routine..... 93

48. The Initial Analysis of the Fourth Query Routine Based on the Number of Relational Operations..... 95

49. The Detailed Analysis of the Fourth Query Routine Based on the Number of Relational Operations..... 95

50. The Comparison of the Number of Relational Operations Between the Normalized Form Schemes During the Fourth Query Routine..... 95

51. The Comparison of the Number of Relational Operations Between Academic Majors During the Fourth Query Routine..... 96

52. The Comparison of the Number of Relational Operations Between the User's Cognitive Style 1 During the Fourth Query Routine..... 96

53. The Comparison of the Number of Relational Operations Between the Academic Majors and the User's Cognitive Style 1 During the Fourth Query Routine..... 97

54. Machine Time Comparison Between the Third and the Fourth Normal Form Schemes During the Fourth Query Routine..... 97

55. The Initial Analysis of the Fourth Query Routine Based on the Execution Time..... 98

56. The Detailed Analysis of the Fourth Query Routine Based on the Execution Time..... 98

57. Time Comparison Between the Normalized Form Schemes During the Fourth Query Routine..... 99

58. Time Comparison Between Academic Majors During the Fourth Query Routine..... 99

59. Time Comparison Between Academic Majors and the Normalized Form Schemes During the Fourth Query Routine..... 99

60. Space Allocated and Used by the Third and the Fourth Normal Schemes During the First Query Routine..... 120

61. Space Allocated and Used by the Third and the Fourth Normal Schemes During the Second Query Routine..... 121

62. Space Allocated and Used by the Third and the Fourth Normal Schemes During the Third Query Routine..... 122

63. Space Allocated and Used by the Third and the Fourth Normal Schemes During the Fourth Query Routine..... 123

64. The Execution Time Used by Both the Third and the Fourth Normal Form Schemes During the First Query Routine..... 125

65. The Execution Time Used by Both the Third and the Fourth Normal Form Schemes During the Second Query Routine..... 126

66. The Execution Time Used by Both the Third and the Fourth Normal Form Schemes During the Third Query Routine..... 127

67. The Execution Time Used by Both the Third and the Fourth Normal Form Schemes During the Fourth Query Routine..... 128

# LIST OF FIGURES


| Figure | | Page |
|---|---|---|
| 1. | Lossless Decomposition | 19 |
| 2. | Data Record Representations | 34 |
| 3. | Functional Dependencies | 39 |
| 4. | Multivalued Dependencies | 40 |
| 5. | Lossless Decomposition From the Third Normal Form to the Fourth Normal Form | 42 |
| 6. | The Third Normal Form Scheme | 66 |
| 7. | Approach One to a Query | 67 |
| 8. | Approach Two to a Query | 68 |
| 9. | The Query Access Paths for the First Query Routine | 113 |
| 10. | The Query Access Paths for the Second Query Routine | 114 |
| 11. | The Query Access Paths for the Third Query Routine | 116 |
| 12. | The Query Access Paths for the Fourth Query Routine | 117 |

# ABSTRACT

The purpose of this thesis is to explore the relationship between the efficiency of the third and the fourth normal forms in a database scheme and to prove the fourth normal form is more efficient and easier to use than the third normal form if the unnormalized form scheme contains a multivalued dependency.

Three methodologies are used to measure the efficiency of the third and the fourth normal form schemes. The first methodology measures the efficiency from the mathematical standpoint. The second methodology measures the efficiency from the machine standpoint, which is the machine time and the memory space required to execute a query. The third methodology measures the efficiency from the end user standpoint which is measured by counting the number of relational operations and execution time needed to execute a query.

First, if multivalued dependencies are present but not detected within the data model, a substantial amount of data redundancy will be expected to occur. If the third normal form scheme is utilized, the mathematical proof has shown that the memory space increases in size in a multiplicative fashion as the complexity of the data relationship increases. If the fourth normal form scheme is utilized, the memory space will increase in an additive fashion as the complexity of the data entities increases.

Second, during the machine efficiency measurements, the fourth normal form scheme performed more efficiently than the third normal form scheme in terms of the memory space utilization and the execution time. These observations were made during the machine simulation that encompassed four query exercises.

Third, observations regarding the relative ease of use associated with a series of end user queries has shown that the fourth normal form data scheme tended to be an easier database scheme to use. Ease of use was measured by the number of relational operations employed and execution time associated with a completed query routine.

Furthermore, the user's cognitive style has been shown to be an important factor in measuring the relative ease of use of a query. The user's cognitive style may become a variable of interest during the database design phase. This should allow the database system to perform more efficiently when designed to incorporate the user's cognitive traits.

The thesis proves that the fourth normal form scheme is relatively more efficient and easier to use than the third normal form in terms of 1) the memory space allocation and utilization of both the theoretical and the machine stand points, 2) the machine execution time, and 3) the end user efficiency measurement measured by the number of relational operations employed and execution time associated with a completed query routine.

# CHAPTER ONE

## INTRODUCTION

The purpose of this research was to explore the relationship between the efficiency of the third and the fourth normal forms in a database scheme. The major hypothesis was that the fourth normal form is more efficient and easier to use than the third normal form if the unnormalized form scheme contains a multivalued dependency.

The third normal form is concerned with single-valued facts. A single-valued fact can be a one-to-one relationship or a one-to-many relationship, whereas the fourth normal form is associated with multivalued facts. A multivalued fact may correspond to a many-to-many relationship among data elements (Kent 1983, 121-123).

A database based on the third normal form is free from the four anomalies derived from functional dependency, where the four anomalies refer to maintenance problems within a relational database scheme. Some data redundancy is observed at this lower level of normalization. A fourth normal form database can sustain a minimal data redundancy by utilizing multivalued dependencies. The main difficulty corresponding to the fourth normal form is to capture an accurate set of multivalued dependencies.

James Martin (1983) stated that using the concept of the fourth normal form has caused confusion in database practice due to the confusion over defining the fourth normal form and the abstruse style of papers on the subject. Furthermore, Martin stated that "In common practice examples of non-fourth-normal-form data are rarely encountered, and easy to avoid" (1983, 233). While James Martin is opposed to idea of the fourth normal form, Daniel Martin reflects his opinion on this subject (1986,41):

> Historically it took many years to study the implications of relations of functional dependence between attributes, which is the opposite of independence. Both within a given relation and between different relations, the semantics of functional dependence interfered with the representation of the relation's data. We shall see that semantic problems occur when manipulating relations with relational algebra and show an example: the connection trap. Today a consensus has been reached: the use of the Fourth Normal Form.

The connection trap refers to a general problem of violation of data integrity and causes the loss of information or creates false information (Daniel Martin 1986, 41). The issue of data independence and integrity will be discussed in Chapter 4.

The motivation of this research is to support the argument for the usage of the fourth normal form with respect to the efficient use of the database. The research verified the hypothesis. The methods used to verify the hypothesis were: (1) theoretical proofs that the fourth normal form allocates less space than the third normal form in most situations; (2) time and space measurements of the third and the fourth normal form schemes done by a computer simulation; and (3) measurements of the number of operations and query formulation time required by users.

The conclusion based on the above methods is as follows: the third normal form with a multivalued dependency will create more data redundancies within a database as the complexity of the database increases. Hereafter, the third normal form refers to the third normal form with a multivalued dependency. The conclusion is restated as follows: The storage taken by a third normal form based database increases multiplicatively along with the complexity of a database. If the third normal form is losslessly decomposed into a fourth normal form based database, then the storage used by the fourth normal form increases additively. Therefore, the fourth normal form data scheme improves data redundancy control, and the efficiency of using a database is also increased.

Chapter 2 deals with the definition of a database system and the relative advantage of using a relational database model. Chapter 3 discusses normalization theory and normalized forms. Chapter 4 discusses the issue of data integrity from the view point of the third and fourth normal data schemes and their difference. In Chapter 4, the difference between the third and fourth normal form is also illustrated through the decomposition scheme with an emphasis on a multivalued dependency. If the third normal form contains a set of multivalued dependencies, the difference is manifested during the normalization.

Chapters 5 and 6 explore the hypothesis from the perspective of computer space utilized by each database scheme. Chapter 5 analyzes the mathematical difference between the two normalized schemes with respect to their storage requirements. Two mathematical proofs are

given. One proof illustrates a multiplicative increase of secondary storage space of the third normal form scheme. The other proof illustrates an additive increase of secondary storage space of the fourth normal form scheme. Chapter 6 discusses the methodology of the machine efficiency measurements and the results. It concludes with the fact that the fourth normal scheme is more efficient than the third normal form scheme with respect to the storage requirement and the execution time.

Chapter 7 analyzes the efficiency measurements between the third and the fourth normal data schemes with respect to the user's perspectives. Based on analyses presented in Chapter 7, three conclusions are reached. One conclusion is that the fourth normal form scheme is found to be more efficient in terms of machine efficiency measurements and the ease of use when compared with the third normal form database scheme. The second conclusion is that computer professionals require less time and a smaller number of commands to complete a query than business users. The third conclusion is that the user's cognitive style must be considered during the database design phase since the user's cognitive style can be a very important factor in determining ease of use. Chapter 8 summarizes and concludes the thesis.

# CHAPTER TWO

# DATABASE MANAGEMENT SYSTEMS

## Introduction

In this chapter, the definition of a database management system (DBMS) and three data models related to database management systems are presented. The advantages of a relational data model are also discussed. There are various approaches to defining a database management system. Definitions of a database seem to differ from author to author, but one objective is consistent: to help a user retrieve data effectively and efficiently. Sample definitions from different authors illustrate the alternative views of a database management system.

Daniel Martin defines a database as "a collection of information on a well-defined subject that is exhaustive, nonredundant, and structured (1986, 5).

The concept of exhaustivity addresses the issue of the completeness of a database. A database must contain all reasonable information about the data in a database. Nonredundancy makes the minimal storage requirements of a database and the enforcement of consistency possible. If a database stores the same information more than once, an inconsistent situation occurs which leads to the poor utilization of memory capacity. With redundant data elements there is

no assurance that updates will correspond to all data entries. Consequently, a query may yield a contradicting result due to the lack of integrity generated via incorrect data updating. Martin, however, noted that "nonredundancy is not always easy to achieve and must sometimes be compromised" (1986, 5-6).

McFadden and Hoffer define a database as follows (1985, 3):

> ... a shared collection of interrelated data designed to meet the varied information needs of an organization. A data base has two important properties: it is integrated and it is shared. By integrated we mean that previously distinct data files have been logically organized to eliminate (or reduce) redundancy and to facilitate data access. By shared we mean that all qualified users in the organization have access to the same data, for use in a variety of activities.

These authors discuss the advantages of a database as compared to traditional file approaches. McFadden and Hoffer identify 10 benefits of the database approach and there are 1) minimal data redundancy, 2)consistency of data, 3) integration of data, 4)sharing of data, 5) enforcement of standards, 6) ease of application development, 7) uniform security, privacy, and integrity controls, 8) data accessibility and responsiveness, 9) data independence, and 10) reduced program maintenance (1985, 15-19).

Martin's database definition is focused on an idea of how information should be stored and structured within a database. McFadden and Hoffer's definition is predicated on how a database should be organized in terms of an organization's and users' views. Both definitions emphasize the requirement that a database management

system (DBMS) should allow the use of data to cross operational, functional, or organizational boundaries. A DBMS, therefore, enables the database users to satisfy multiple "views" of the data stored.

## Fundamental Database Models

When a database is implemented, the database system usually is structured for using one of three models which are hierarchical, network, or relational in nature. Each model has its own definition and characteristics. A brief presentation of each model follows:

### Hierarchical Database Model

The hierarchical data model has the following characteristics:

A hierarchical schema consists of two sets:

(i) A set of entity types E. Particular entity types will be denoted as E1, E2, . . . , En.

(ii) A set of logical relationships L among entity types. The relationship between entity types Ei and Ej will be denoted as Lij.

The logical relationships among entity types satisfy the following requirements:

(iii) There exists at most one relationship Lij between different entity types Ei and Ej. As a consequence, there is no need to name the relationships.

(iv) There exists no relationship between an entity Ei and itself; that is, Lii is not a member of the set L (Alagic 1986, 49).

Furthermore Alagic (1986, 53) states that "An entity type may be the owner in one or more sets, but as a member it can participate in only one set." McFadden and Hoffer state that

> The hierarchical data model represents data as a set of nested one-to-many (1:M) and one-to-one (1:1) relationships. ... Because of the similarity of hierarchies and tree data structures, terminology like root, level, and leaf are also used when discussing hierarchies (McFadden and Hoffer 1985, 169).

Atre (1980) discusses advantages and disadvantages of a hierarchical data model. The advantages of a hierarchical scheme are as follows:

> The major advantage of the hierarchical data model is the existence of proven data base management systems that use the hierarchical data model as the basic structure.
> The relative simplicity and ease of use of the hierarchical data model and the familiarity of data processing users with a hierarchy are major advantages.
> There is a reduction of data dependency. ...
> Performance prediction is simplified through predefined relationships (Atre 1980, 106).

Note that the hierarchical model has the following disadvantages:

> The many-to-many relationship can be implemented only in a clumsy way. This may result in redundancy in stored data. ...
> As a result of the strict hierarchical ordering, the operations of insertion and deletion become unduly complex.
> Deletion of parent results in the deletion of children. ...
> Hierarchical commands tend to be procedural because of the strictness of the structure.
> "Root" is the dominant node type. Any child node is accessible only through its parent node (Atre 1980, 106-109).

Network Database Model

The network model was developed to compensate for the hierarchical data model's limitations associated with data access and retrieval. The network model has the following characteristics:

> (1) Sets
> If E1 and E2 are entity types such that there exists a functional relationship E2 -> E1, then the ordered named pair E(E1, E2) is called a set. E1 is called the owner of the set E and E2 the member of the set E. ...
> (3) Set Occurrences
> An occurrence of a set consists of an occurrence of the owner entity type E and zero or more occurrences of the member entity type E. In terms of functional relationships this means that a set occurrence consists of a set of member entities and the owner entity to which they are mapped according to the functional relationships. ...

(4) Uniqueness of Owner
The fact that a set is in fact a representation of a functional relationship may be expressed by the condition that an entity may participate as a member in only one occurrence of a given set. Because of that, a member entity has a unique owner entity in every set in which it participates as a member (Alagic 1986, 51-55).

Daniel Martin (1986, 58) states that

> The network model features N-to-P relationships. Each N-to-P relationship usually consists of two 1-to-N relationships in opposite directions linking the same pair of records.

The advantages of the network model relative to the hierarchical form are as follows:

> The major advantage of the network data model is that, as for the hierarchical data model, there are successful data base management systems that use the network data model as the basic structure.
>
> The many-to-many relationship, which occurs quite frequently in real life, can be implemented easily.
>
> The network data model is backed by the CODASYL (Conference On Data Systems Languages) Data Base Task Group (DBTG) (Atre 1980, 121).

Atre states the disadvantages of using the network model (1980, 121):

> The main disadvantage of the network model is its complexity. The application programmer must be familiar with the logical structure of the data base, because she/he has to "navigate" through different set occurrences with the help of connector record type occurrences. ...
> ... it is possible, unless great care is taken, to lose data independence.

## Relational Database Model

The third database model is the relational model, which has the following characteristics:

> A relation scheme R is a finite set of attribute names {A1,A2,...,An}. Corresponding to each attribute name Ai is a set of Di, 1 ≤ i ≤ n, called the domain of Ai. We also

> denote the domain of Ai by dom(Ai). Attribute names are sometimes called attribute symbols or simply attributes, particularly in the abstract. The domains are arbitrary, non-empty sets, finite or countably infinite. Let D = D1 U D2 U ... U Dn. A relation r on relation scheme R is a finite set of mappings {t1,t2,...,tp} from R to D with the restriction that for each mapping t ∈ r, t(Ai) must be in Di, 1 ≤ i ≤ n. The mappings are called tuples. ...
> ... a tuple is a set of values, one for each attribute name in the relational scheme. ...
> A key of a relation r on relation scheme R is a subset K = {B1,B2,...,Bn} of R with the following property. For any two distinct tuples t1 and t2 in r, there is a B K such that t1(B) = t2(B) (Maier 1983, 2-4).

There are two formal languages for the relational database in order to represent queries about properties of entities represented by the relational database: relational algebra and relational calculus.

> Relational algebra manipulates one or two relations as operands and produces a new relation as the result. ... Relational calculus manipulates relations implicitly by specifying conditions that can involve attributes from several relations (McFadden and Hoffer 1985, 191-193).

The four criteria of the normalization process, mentioned above, are discussed in detail in Chapter 3.

The relational data model offers the following advantages over nonrelational data forms: (1) simplicity, (2) nonprocedural requests, (3) data independence, and (4) theoretical foundation (Atre 1980, 94-95). Each advantage will be discussed in the context of a comparison between a relational model and a nonrelational model. According to Daniel Martin (1986, 54) the primary difference between a relational database and nonrelational (navigational) database is that a relational database manipulates one relation at a time, whereas a

nonrelational database manipulates one record at a time. A nonrelational database may require a program to achieve the same result which a relational database can do with one operation.

A nonrelational database is also called a navigational database since "... navigational means 'what describes which path to use to find the data in the database' " (Martin 1986, 54). A nonrelational, navigational, database must follow a path navigated by a program, whereas a relational database needs to know what to obtain from the data entity. The database management system requires

> ... the addition of logic to navigate the DBMS structures to derive the desired application structures.
>
> Such applications suffer from significant problems:
> - They tend to be larger and more difficult to write
> - The potential for multiple errors is great
> - Development time tends to be expanded
> - Applications therefore require larger resource investments
> - Changes in the database structure may require reinvestment in applications to enable them to derive the same desired data structures
> - Applications can only depend on the DBMS for management of the files over which it has control (Wood 1985, 15).

The final difference between a relational and nonrelational database is the type of tools available for each database in order to formulate complex queries. A relational database utilizes relational algebra and relational calculus, which are derived from set theory; however, a nonrelational database utilizes a data manipulation language. A typical data manipulation language has more than 200 reserved words; however, commonly used relational databases use only approximately 24 reserved words (Martin 1986, 55).

McFadden and Hoffer discuss five fundamental differences between relational and nonrelational database in addition to Martin's observation. McFadden and Hoffer listed these differences to illustrate the advantages of using a relational database (1985, 183-184):

1. Implementation independence: the relational model logically represents all relationships implicitly, hence, one does not need to know what association are or are not physically represented by an efficient access path (without looking at the internal data model).
2. Terminology: the relational model has been developed with its own set of terminology, most of which has equivalent terms in other data models.
3. Logical key pointers: the relational data model uses primary (and secondary) keys in records to represent the association between two records; ...
4. Normalization theory: properties of a data base that make it free of certain maintenance problems have been developed within the context of the relational model....
5. High level programming languages: programming languages have been developed specifically to access data bases defined via the relational data model; these languages permit data to be manipulated as groups or files and not procedurally one record at a time.

All three authors believe that a relational model is more powerful than the other database models because of the above observations.

The advantages of using a relational model are summarized, based on all three authors, as follows:

1) how the database executes a query;
2) tools available to manipulate data items; and
3) a well defined set of mathematical terminology (Martin 1986; McFadden and Hoffer 1985).

Farin and Nazario (1986, 46) state the disadvantages of using a relational model.

> The main weakness of a relational DBMS stems from its strengths. There is a potential risk of costly and inefficient access. Users may not select the optimal access path, and may not state their data requests in the manner most efficient for the DBMS. ...
> It is also difficult for a data base designer to optimize the physical representation of data prior to its use. These problems become more readily apparent when large databases are being maintained.

The issue of user efficiency with regard to relational data base design is one major focus of this thesis and will be discussed in Chapter 7.

# CHAPTER THREE

# NORMALIZATION THEORY AND NORMAL FORMS

## Introduction

In this chapter, three concepts related to the relational database model are discussed. The first concept is the normalization theory, which guides the database designer in the construction of a database. The second concept is the dependency theory, which enables the database designer to justify the degree of normalization. The third concept is that of a normal form, which is the end product of the normalization process.

## Normalization Theory

McFadden and Hoffer define normalization and the purpose of normalization as follows (1985, 221):

> Normalization is the analysis of functional dependencies between attributes (or data items). The purpose of normalization is to reduce complex user views to a set of small, stable data structures. Experience clearly shows that normalized data structures are more flexible, stable, and easier to maintain than unnormalized structures.

A possible disadvantage to the normalization process seems to exist. For instance, assume finding a certain piece of information within a database is necessary. When a normalized form is used, often a costly join operation must be utilized. As a result, a new relational table is created. Then the selection of a data field from the newly created

relation must be made. However, in order to extract the same information from an unnormalized form, a simple selection and projection operation is required. One must carefully define the benefits of data normalization in order to fully appreciate the power of a normalized relational database model.

Perkinson (1984, 42) discusses two benefits from database normalization: the minimization of "data maintenance" and minimization of "program maintenance." "Maintenance involves both a minor task such as expanding the length of a field and complex tasks such as adding new fields due to increased functional complexity" (Perkinson 1984, 41). Minimization of data maintenance is concerned with the problem of maintaining integrity during data update. Whenever a data item is modified, that item must be changed every time it appears in the database, either directly by the user, or indirectly by the system.

Minimization of program maintenance deals with a problem of internal change inside of the database. Perkinson (1984) gave an example of expanding the length of a certain data field. This internal change requires the database to locate and update every program that is used in the database, even for programs that do not include a specified data field. Only programs which "contain the data in question should be updated" (Perkinson 1984, 41).

Without the concept of normalization, a possible approach to avoiding the problems of maintenance is to break down the original large data file into a set of smaller data files. Each data item is related to every other item within a data file. However, this is an

intuitive notion of the normalization process. Decomposition of a large file into a set of small manageable files is the fundamental idea of normalization.

Four Criteria of Normalization

The four criteria of normalization as discussed by Ullman are summarized into four components (1982, 212):

1) Redundancy
2) Potential inconsistency (update anomalies)
3) Insertion anomalies
4) Deletion anomalies

These four criteria are also known as maintenance problems within the relational database scheme. These four anomalies and their elimination are used to introduce the existence of different levels of normal forms. A lower normal form suffers from these four anomaly situations, whereas a higher normal form solves them. These four anomalies are called the criteria of normalization since they are a means of progressing from a lower normal form to a higher normal form.

Data redundancy is a situation where the logical database scheme allows a particular data element to be stored in more than one location within the database. The logical database scheme may suffer from the following problems if data redundancy is allowed to occur:

1) Storage space in the computer may be wasted due to data redundancy;
2) In order to update all occurrences of a particular data item, several input prcedures must be made directly or indirectly by a user; and
3) Inconsistencies within the database may be exhibited if all occurrences of the same data items are not inserted (deleted, or updated) (McFadden and Hoffer 1985, 11).

An insertion anomaly is observed when a user inserts a new data item without inserting a value for every primary key. If an undefined value for a primary key is allowed within a database, it may lead to an update anomaly or the destruction of the functional dependencies among the data items. Thus, every primary key must have a defined value within the database. A primary key refers to "a key that is used to uniquely identify a record instance" (Martin 1983, 746).

A deletion anomaly occurs when a user fails to delete a data item from the database when a primary key has been deleted; conversely, when a user tries to delete a particular data item, some important but unrelated information may be inadvertently deleted.

An update anomaly is observed when a user fails to update the database properly. When multiple occurrences of data items in a data file are permitted, updating only a single occurrence of the data items may lead to an inconsistency in the database. Duplicate data items may not be updated simultaneously.

## Dependencies and Redundancies

The relationship between dependencies and redundancies is the key to the normalization process. Avoidance of redundancies forces an original data file to be decomposed into a set of small data files. The rules utilized to decompose an original data file into a set of smaller data files are called dependencies.

> ... given the fact that a relation R is legal, i.e., satisfies certain dependencies, and given certain information about the value of R, we should be able to deduce other things about R (Ullman 1982, 213).

The dependency not only causes the redundancies, but also permits a lossless decomposition. An example can be used to illustrate this concept. Assume we have a relation SUPPLIER(SNAME, SADDRESS, ITEM, PRICE). The four maintenance problems mentioned earlier are observed in this relation. Since data redundancy illustrates the need for detection of a dependency, data redundancy is focused on. In this example, redundancy is easily detected since "the address of the supplier is repeated once for each item supplied" (Ullman 1982, 212). If the content of SNAME is known, it is easy to deduce the content of SADDRESS. (SADDRESS is functionally dependent on SNAME.) All but one attribute of SADDRESS in this relation is redundant. In order to avoid any redundancy, one can break down SUPPLIER into two distinct relations, namely SA(SNAME, SADDRESS) and SIP(SNAME, ITEM, PRICE) (Ullman 1982, 212). Furthermore, one can recover the original relation by joining SA and SIP by using a natural join. Therefore we have achieved the lossless decomposition (see Figure 1).

Several dependencies are central to the theory of normalization. Three dependencies are emphasized for normalization processes: (1) functional dependencies, (2) multivalued dependencies, and (3) join dependencies. These dependencies will be discussed in the sections which follow.

SUPPLIER(SNAME, SADDRESS, ITEM, PRICE)

|
|Normalization is undertaken.
V

SA(SNAME, SADDRESS)

and

SIP(SNAME, ITEM, PRICE).

SA(SNAME, SADDRESS)  SIP(SNAME, ITEM, PRICE)

|
|Natural join is undertaken.
V

SUPPLIER(SNAME, SADDRESS, ITEM, PRICE)

Figure 1. Lossless Decomposition.

Functional Dependency. The definition of functional dependency (FD) is as follows:

> Let R(A1,A2,...,An) be a relation scheme, and let X and Y be subsets of{A1,A2,...,An}. We say X -> Y, read 'X functionally determines Y' or 'Y functionally depends on X' if whatever relation r is the current value for R, it is not possible that r has two tuples that agree in the components for all attributes in the set X yet disagree in one or more components for attributes in the set Y (Ullman 1982, 214).

Therefore, two tuples that agreed on the attributes of X would have to be the same entity. Ullman further states that:

> ... if R represents a many-one mapping from entity set E1 to entity set E2, and among the Ai's are attributes that form a key X for E1 and a key Y for E2, then X -> Y would hold, and in fact, X functionally determines any set of attributes of R. However, Y -> X would not hold unless the mapping were one-to-one (1982, 214).

Functional dependency is limited to representing one-to-one and one-to-many relationships. An illustration of functional dependencies may help to visualize this concept more clearly. A functional

dependency is found in the relation SA: SNAME -> SADDRESS as was discussed earlier. If one knows the name of the supplier, one can find the address since SNAME functionally determines SADDRESS. By asserting that a functional dependency exists, the storage of certain information can be limited. If a functional dependency: SNAME -> SADDRESS holds, no supplier can have two addresses in the database system (Ullman 1982, 215). If in actuality more than one address exists, the problem can be resolved by using multivalued dependencies.

Multivalued Dependency. The concept of multivalued dependency (MVD) has been introduced by Ronald Fagin (1977, 262-282). Fagin called a generalization of functional dependencies a multivalued dependency. Maier defines multivalued dependency as follows:

> Let R be a relation scheme, let X and Y be disjoint subsets of R, and let Z = R - (X Y). A relation r(R) satisfies the multivalued dependency (MVD) X ->-> Y if, for any two tuples t1 and t2 in r with t1(X) = t2(X), there exists a tuple t3 in r with t3(X) = t1(X), t3(Y) = t1(Y), and t3(Z) = t2(Z).
>
> The symmetry of t1 and t2 in this definition implies there is also a tuple t4 in r with t4(X) = t1(X), t4(Y) = t2(Y) and t4(Z) = t1(Z) (1983, 124).

Maier also states the relationship between multivalued dependencies and functional dependencies: "Let r be a relation on scheme R and let X and Y be subsets of R. If r satisfies the FD X -> Y, then r satisfies the MVD X ->-> Y" (1983, 127). Therefore, multivalued dependency is merely a generalization of functional dependency. To illustrate the need for multivalued dependencies, assume the following example: A relation R(CLASS, SECTION, STUDENT, EXAM, TEXT) (Fagin 1977, 268-270). There are no functional dependencies among this

relation; therefore, this relation is free from any of the four anomalies from the perspective of functional dependencies. However, this relation has the following property: if a particular class and section used more than one textbook and has more than one examination, an examination is dependent only on the class and section and not on the textbook used. In other words, for each class section, there is a well defined set of textbooks and a set of examinations. The set of textbooks is independent from the set of examinations but is dependent on class section alone.

Despite the fact that this relation does not have any functional dependencies, there exists a great deal of data redundancy. This data redundancy leads to the creation of the other three anomalies. For example, if a particular class section decides to drop a textbook, several deletions must be made (as many as the number of examinations in a particular class section). If a new textbook is added to a certain class section, several tuples must be created. Due to this multivalued dependency in the relation (CLASS SECTION ->-> TEXT, CLASS and SECTION STUDENT ->-> EXAM ), one can further decompose the relation. Two projections R1 and R2 are the results of decomposition. R1 contains CLASS, SECTION, STUDENT, EXAM, and R2 contains CLASS, SECTION, TEXT. This projection now sets the database free from any of the four anomalies.

Join Dependency. The last dependency used in normalization theory is called join dependency. The concept of join dependency was first

introduced by Rissanen (1977, 317-325). Maier states the concept behind join dependencies (1983, 139):

> MVDs are an attempt to detect lossless decompositions that will work for all relations on a given relation scheme. However, MVDs are not completely adequate in this regard. A relation can have a nontrivial lossless decomposition onto three schemes, but have no such decomposition onto any pair of schemes....

Date (1986) clarified the join dependency theory by using the idea of n decomposability. As far as the fourth normal form is concerned, replacing a relation by two of its projections is the sole operation necessary or available in the nonlossless decomposition process (Date 1986, 385). A projection refers to the component of the database after the decomposition process is undertaken. There exist relations in which nonlossless decomposition can be achieved only by breaking down the relation into three or more projections, (i.e., two projections are not enough to achieve lossless decomposition.) A definition of join dependencies is as follows:

> Let R = {R1,R2,...,Rp} be a set of relation schemes over U. A relation r(U) satisfies the join dependency (JD) *[R1,R2,...,Rp] if r decomposes losslessly onto R1,R2,...,Rp. That is, r = $\pi$R1(r) $\bowtie$ $\pi$R2(r) $\bowtie$ ... $\bowtie$ $\pi$Rp(r). We also write *[R1,R2,...,Rp] as *[R] (Maier 1983, 139-140).

Date (1986, 388) states that "JDs are a generalization of MVDs" and that they are "the most general form of dependency possible." To illustrate a join dependency, assume there exists a relation R(AGENT, COMPANY, PRODUCT). Furthermore there are no multivalued dependencies in R; however, a join dependency exists: *{(AGENT, COMPANY), (AGENT, PRODUCT), (COMPANY, PRODUCT)}. Due to the join dependency, one can decompose R into three projections: R1 = (AGENT, COMPANY), R2 =

(AGENT, PRODUCT), and R3 = (COMPANY, PRODUCT). If one arbitrarily chooses to join two projections from these three relations, a superfluous relation is created. In order to recover the original relation R, a natural join of all the three projections must be undertaken (Kent 1983, 123-125).

To summarize the three proceeding sections, normalization is the key process in the database design. In order to achieve lossless decomposition, a database designer must always confront the four criteria associated with normalization and dependencies. The focus of discussion will now shift to normal forms, an end product of normalization. Normal forms are used to represent a conceptual database model as the database design process continues to refine the conceptual view.

## Normal Forms

Date (1986, 362) states that "Normalization theory is built around the concept of normal forms. A relation is said to be in a particular normal form if it satisfies a certain specified set of constraints." The concept of normal forms is essential in the relational database structure since a normal form is based upon the relationship existing within a database. There are six normal forms: the first normal form to the fourth normal form, the fifth normal form (also called Project-Join), and the Boyce-Codd normal form. Each normal form will be introduced and its role in a relational database will be discussed below.

The definition of a relational database scheme is stated mathematically according to Maier (1983, 94).

> Let U be a set of attributes, each with an associated domain. A relational database scheme R over U is a collection of relation schemes {R1,R2,...,Rp}, where Ri = (Si,Ki), $1 \leq i \leq p$,
>
> $$\bigcup_{i=1}^{p} Si = U,$$
>
> and $Si \neq Sj$ if $i \neq j$.
> A relational database d on database scheme R is a collection of relations {r1,r2,...,rp} such that for each relation scheme R = (S,K) in R there is a relation r in d such that r is a relation on S that satisfies every key in K. We abuse set notation and always assume that ri is the relation on Si.

S represents a set of attributes and K represents a set of designated keys.

An unnormalized data form is defined as the following:

> An unnormalized relation is a relation that contains one or more repeating groups. As a result, there are multiple values at the intersection of certain rows and columns (McFadden and Hoffer 1985, 222).

The disadvantage of an unnormalized form is that data redundancy is found. The goal of data normalization is to divide a database into a flexible and stable set of simple data schemes while free from the four anomalies previously discussed.

A definition of first normal form is the following:

> A relation scheme R is in first normal form (1NF) if the values in dom(A) are atomic for every attribute A in R. That is, the values in the domain are not lists or sets of values or composite values. A database scheme R is in first normal form if every relation scheme in R is in first normal form (Maier 1983, 96).

The advantage of 1NF is that "it may be not possible to express FDs" without 1NF (Maier 1986, 97). If a list or set of values or

composite values is allowed, there could exist some situation where semantically the same information is not allowed to represent the same functional dependencies. The example used by Maier is FD: "BIRTHDATE -> SIGN, which would allow two people born on the same day in different years to have different signs" (1983, 97).

An example by Date (1986, 367-374) illustrates second and third normal forms with respect to the four criteria of normalization; it will be presented at length below. This relation appears as follows: FIRST(S#, STATUS, CITY, P#, QTY), where S# represents the supplier number, STATUS represents the status of the city, CITY represents the name of the CITY, P# represents the parts number, and QTY represents quantity of the particular part number.

One must apply four criteria of normalization in order to improve the relation FIRST. Data redundancy is detected easily since every tuple with a particular S# must have the same STATUS, and CITY for every different part and the quantity a supplier supplies. An insertion anomaly is observed when a particular supplier in a particular city without a specific part is inserted. According to Date, no component of a primary key value can be unassigned (1986, 367). In relation FIRST, the combination of S# and P# is the primary key. A deletion anomaly is detected when a particular supplier is deleted from the relation FIRST. Not only may a supplier number be deleted but also city information may be deleted as well. An update anomaly is observed when a particular supplier moves to a new city. Two possibilities exist. First, the user searches every tuple with a particular supplier and the previous city name is changed to a new

city name. Second, if all tuples are not changed, an inconsistent result is produced.

In order to fulfill the four criteria of normalization the data structure must be decomposed into the two distinct relations, SECOND(S#, STATUS, CITY) and SP(S#, P#, QTY). Maier defined the second normal form as follows (1983, 99):

> Given a relation scheme R, an attribute A in R, and a set of FDs F over R, attribute A is prime in R with respect to F if A is contained in some key of R. Otherwise A is nonprime in R. ... A relation scheme R is in second normal form (2NF) with respect to a set of FDs F if it is in 1NF and every nonprime attribute is fully dependent on every key of R. A database scheme R is in second normal form with respect to F if every relation scheme R in R is in 2NF with respect to F.

Relation SECOND and SP are both in 2NF since nonkey attributes, STATUS and QTY are fully functionally dependent on the primary key S#, and S# and P# respectively. The reduction process from 1NF to 2NF is achieved by applying suitable projections. The original relation, FIRST, can always be recovered by utilizing the natural join of these projections, SECOND and SP; therefore, no information is lost in this reduction process. Date calls this process a "nonloss decomposition." A nonloss decomposition refers to a decomposition scheme which produces a set of projections and by joining a set of projections no information is lost (Date 1986, 371).

Relation SECOND still does not totally meet the four criteria of normalization. Data redundancy is observed since every city has only one status value. If more than one supplier resides in the same city, information about city status is redundant. An insertion anomaly is encountered when an insertion of a new city is made, particularly when

a supplier exists in a new city and no appropriate primary key value exists for that supplier. A deletion anomaly is observed when a deletion of a particular city is made: not only is the supplier's information deleted but also the status of the city is deleted if there is only one supplier in the city. Observation of an update anomaly is made when changing the status for a particular city. There are two possible solutions available in regard to update anomaly. One possibility is simply to search every occurrence of a tuple with a particular city and change the status. The other possibility is to let an inconsistent result propagate within a database because only one data record is updated (Date 1986, 372).

The solution to these problems is to break down the relation SECOND into two projections, SC(S#,CITY) and CS(CITY,STATUS). These two projections are said to be in the third normal form. Maier defines third normal form as follows (1983, 100):

> Given a relation scheme R, a subset X of R, an attribute A in R, and a set of FDs F, A is transitively dependent upon X in R if there is a subset Y of R with X -> Y, Y $\not\rightarrow$ X, and Y -> A under F and A $\notin$ XY. ... A relation scheme R is in third normal form (3NF) with respect to a set of FDs F if it is in 1NF and no nonprime attribute in R is transitively dependent upon a key of R. A database scheme R is in third normal form with respect to F if every relation scheme R in R is in third normal form with respect to F.

Relations CS, SC, and SP are third normal form since no transitive dependencies exist on nonprime attributes on keys. Furthermore, these three relations are free from the four anomalies.

3NF has certain inadequancies in

the case of a relation that
1. had multiple candidate keys, where
2. those candidate keys were composite, and
3. the candidate keys overlapped (i.e., had at least one attribute in common) (Date 1986, 374).

A candidate key refers to "a key that uniquely identifies normalized record instances of a given type" (Martin 1983, 738).

In order to prevent these design inadequancies, a different normal form is needed to address the four criteria of normalization. To illustrate the situation, a new relation SSP(S#, SNAME, P#, QTY) is introduced and "the candidate keys are (S#, P#) and (SNAME, P#)" (Date 1986, 376). One must note that SSP is in 3NF since the definition of 3NF does "not require an attribute to be fully dependent on the primary key if it was itself a component of some attribute key of the relation" (Date 1986, 376). Therefore, SNAME is not fully dependent on (S#, P#) and was ignored. However, SSP has failed to meet the four criteria of normalization. Data redundancy is easily observed in that every tuple has the S# and SNAME, whereas "S# and SNAME are determinants because each of them determines the other" (Date 1983, 376). An insertion anomaly is detected when a particular supplier is inserted without inserting a corresponding part number into the database. The same logic used to illustrate the second normal form is applied here since no primary key can contain a null value. A deletion anomaly is detected when a particular supplier number is deleted, as well as the other information concerning the supplier. An update anomaly is found when changing the name of a supplier. There are two possibilities: one is to search and change every occurrence

of a particular supplier, and the other is to produce a possibly inconsistent result.

The solution is to decompose the relation SSP into two projections, SS(S#, SNAME) and SP(S#, P#, QTY). Maier defined the new relation as follows (1983, 118):

> A relation scheme R is in Boyce-Codd normal form (BCNF) with respect to a set of FDs F if it is in 1NF and no attribute in R is transitively dependent upon any key of R. A database scheme R is in Boyce-Codd normal form with respect to a set of FDs F if every relation scheme R in R is in Boyce-Codd normal form with respect to F.

The theoretical difference between 3NF and BCNF is that in 3NF only nonprime attributes in R cannot be transitively dependent upon a key of R, whereas in BCNF, any attribute in R cannot be transitively dependent upon a key of R. BCNF mainly calls attention to functional dependencies, which are useful with respect to 1-to-1, many-to-1, and 1-to-many relationships. Functional dependencies alone cannot provide a complete model. There are some situations where no further decomposition can be made with respect to functional dependencies. Thus, a relation fails to meet the four criteria of normalization. If this situation occurs, introduction of a new normal form cannot be avoided. This normal form must satisfy the four criteria of normalization. The new normal form was called a fourth normal form and was first introduced by Fagin (1977, 262, 278).

To depict the difference between the third normal form and the fourth normal form, a relation R(CLASS, SECTION, STUDENT, EXAM, TEXT) will be utilized from Fagin (1977, 267). It is readily shown that relation R fails to meet the four criteria of normalization. Data

redundancy exists within relation R since every occurrence of a data element of EXAM, must accompany all occurrences of data elements of TEXT. It is also the case that every occurrence of a data element of TEXT must accompany all occurrences of data elements of EXAM. There will be M entries of the same information regarding textbooks (TEXT) for every occurrence of examination (EXAM) where M represents the number of distinct textbooks (TEXT). Data redundancy exists since multivalued dependencies exist in this relation: {CLASS, SECTION} ->-> TEXT and {CLASS, SECTION, STUDENT} ->-> EXAM. However, the relation R does not have any determinants. Therefore, the relation R is in the third normal form.

The relation R fails to satisfy the four criteria of normalization. An insertion anomaly is detected when an insertion of a new textbook is made without an entry of examination type. This insertion is not allowed since CLASS, SECTION, STUDENT, EXAM, and TEXT is the only key (composite key) in relation R, and no key attribute can have a null value associated with it. A deletion anomaly is observed when the only exam score in the relation is deleted. At the same time the other information about CLASS, SECTION, STUDENT, and TEXT is deleted. An update anomaly is detected when a certain updating mechanism is performed. Every tuple must be searched and updated accordingly. Otherwise, an inconsistent result may be produced.

To resolve these four anomalies, the lossless decomposition of relation R into two its projections, R1={CLASS, SECTION, TEXT} and R2={CLASS, SECTION, STUDENT, EXAM} must be done. These two

projections are called fourth normal forms. Maier defined the fourth normal form as follows (1983, 136):

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

A superkey is a broad definition of a key, similar to the definition provided in Chapter 2 (Maier 1983, 4-5).

Finally, the fifth normal form is introduced. Date states that "the 5NF is the ultimate form" but that fifth normal forms "are pathological cases and likely to be rare in practice" (1986, 390). Maier defined the fifth (project join) normal form as follows (1983, 141):

> Let R be a relation scheme and let F be a set of FDs and JDs. R is in projection-join normal form (PJNF) with respect to F if for every JD *[R1,R2,...,Rp] implied by F that applies to R, *[R1,R2,...,Rp] is implied by the key FD of R.

To summarize, there is a well defined theory of normalization to support relational databases. The three most important forms are as follows: the third, the Boyce-Codd, and the fourth normal forms, where the third and the Boyce-Codd normal forms deal with single These normal forms are practiced widely depending upon the application. The following chapter deals with third and fourth normal forms, since the efficiency measurement of both normalized forms is the topic of this research.

# CHAPTER FOUR

# DATA INTEGRITY OF THIRD AND FOURTH NORMAL FORMS

## Introduction

In this chapter, the issue of data integrity in third and fourth normal form schemes is discussed. First, a definition of data integrity is presented and how both normalized forms played their roles in the issue of data integrity. Second, three maintenance policies, which maintain data integrity within the third normal form scheme, are discussed. These maintenance policies become a crucial issue when the third normal form contains the multivalued dependency which is considered a violation of the fourth normal form. Third, data independence is discussed as a problem of detecting a multivalued dependency. Fourth, a difference between the third and the fourth normal forms is illustrated from the logical view.

## Data Integrity

One of the goals of a database management system is data integrity, that is, maintaining the validity of data. Often,

maintaining data integrity can be expensive and difficult to achieve. Normalization is the technique that allows a database management system to sustain data integrity. The third and fourth normal forms were chosen since these forms are widely used during the database system design step.

The third normal form is concerned with single-valued facts. A single-valued fact can be a one-to-one relationship or a one-to-many relationship, whereas the fourth normal form is associated with multivalued facts. A multivalued fact may correspond to a many-to-many relationship among data elements (Kent 1983, 121-123).

When facing a many-to-many relationship among data entities, there are two ways to normalize a data record. One way is to use the third normal form, which utilizes the single data record approach. The other method is to use the fourth normal form, which involves breaking down a data entity into two data records. A record appears in the third normal form "if every field is either part of the key or provides a (single-valued) fact about exactly the whole key and nothing else" (Kent 1983, 121). "Under fourth normal form, a record type should not contain two or more independent multivalued facts about an entity" (Kent 1983, 122). Independence is a very important issue and is discussed later in this chapter.

## Maintenance Policies

A maintenance policy becomes crucial for data integrity if a relation in the third normal form contains a multivalued dependency

which is considered a violation of the fourth normal form. This section discusses the possible approaches used in defining various maintenance policies and the selection of the most appropriate maintenance policy. The representation method for a third normal form projection within the database is also introduced.

Consider the following example, extracted from Kent (1983, 122), to illustrate the maintenance policies necessary when violating the fourth normal form. Assume that we have employees within a firm that have different skills and speak various languages. Two many-to-many relationships can be discovered, the first between employees and skills, and the second between employees and languages (Kent 1983, 122). "Under fourth normal form, these two relationships should not be represented in a single data record .... Instead, they should be represented in the two records .... (see Figure 2) (Kent 1983, 122).

EMPLOYEE SKILL LANGUAGE

KEY

A single record representation

EMPLOYEE SKILL

KEY

EMPLOYEE LANAGUAGE

KEY

A two record representation

Figure 2. Data Record Representations.

Source: Kent, W. "A Simple Guide To Five Normal Forms in Relational Database Theory." Communication of ACM 26, no. 2. (1986): 122.

Kent (1983, 122) further discusses that there are three maintenance policies if the fourth normal form is violated, since the violation of the fourth normal form leads to uncertainties as to how to store data records. The three maintenance policies can be used to maintain "two independent multivalued facts in one record" (Kent 1983, 122).

Disjoint Format Policy. The first maintenance policy is called a disjoint format. This format allows a record to be stored as either a skill or a language, but not as both (see Table 1).

Table 1. The Disjoint Format Maintenance Policy.

| EMPLOYEE | SKILL | LANGUAGE |
|---|---|---|
| Smith | cook | |
| Smith | type | |
| Smith | | French |
| Smith | | German |
| Smith | | Greek |

Source: Kent, W. "A Simple Guide To Five Normal Forms in Relational Database Theory." Communication of ACM 26, no. 2. (1986): 122.

The problems associated with the first policy are:

> This is not much different from maintaining two separate record types. We note in passing that such a format also leads to ambiguities regarding the meanings of blank fields. A blank SKILL could mean the person has no skill, that the field is not applicable to this employee, that the data is unknown, or, as in this case, that the data may be found in another record (Kent 1986, 122).

Random Mix Policy. The second maintenance policy is called a random mix. There are three variations of a random mix. The first variation

is a "minimal number of records with repetitions", the second is a "minimal number of records with null values", and the third is "unrestricted" record fields (see Table 2) (Kent 1983, 122).

The problems associated with the second policy are that it is possible to convey wrong information. In addition, if one of the data records is deleted, it is possible to lose a specific piece of information about the employee. Finally, a blank field may be open to various interpretations.

Table 2. The Random Mix Maintenance Policy.

(1) Minimal number of records with repetitions

| EMPLOYEE | SKILL | LANGUAGE |
|---|---|---|
| Smith | cook | French |
| Smith | type | German |
| Smith | type | Greek |

(2) Minimal number of record with null values

| EMPLOYEE | SKILL | LANGUAGE |
|---|---|---|
| Smith | cook | French |
| Smith | type | German |
| Smith | | Greek |

(3) Unrestricted

| EMPLOYEE | SKILL | LANGUAGE |
|---|---|---|
| Smith | | French |
| Smith | cook | |
| Smith | cook | German |
| Smith | type | Greek |

Source: Kent, W. "A Simple Guide To Five Normal Forms in Relational Database Theory." Communication of ACM 26, no. 2. (1986): 122.

Cross Product Policy. The third maintenance policy is called a cross product form (see Table 3). This maintenance policy requires

every employee to have "a record for every possible pairing of one of his skills with one of his languages" (Kent 1986, 122).

Table 3. The Cross Product Maintenance Policy.

| EMPLOYEE | SKILL | LANGUAGE |
|---|---|---|
| Smith | cook | French |
| Smith | cook | German |
| Smith | cook | Greek |
| Smith | type | French |
| Smith | type | German |
| Smith | type | Greek |

Source: Kent, W. "A Simple Guide To Five Normal Forms in Relational Database Theory." Communication of ACM 26, no. 2. (1986): 122.

The problem associated with the cross product form is that "updates have to be done in multiple records" (Kent 1986, 123).

> ... we mention that fourth normal form is defined in terms of multivalued dependencies that correspond to our independent multivalued facts. Multivalued dependencies, in turn, are defined essentially as relationships that accept the 'cross-product' maintenance policy mentioned above. For our example, every one of an employee's skills must appear paired with every one of his languages. It may or may not be obvious to the reader that this is equivalent to our notion of independence; since every possible pairing must be present, there is no 'information' in the pairings. ... If all pairings are always present, then the relationships are really independent (Kent 1983, 123).

## Data Independence

The issue of independence becomes crucial since the fact that data independence often limits the higher normalization of third

normal form. In this section, a definition of data independence and a limit of data independence are discussed.

Independence among the many-to-many relationship results from the fact that there is no direct connection between entities.

> The two many-to-many relationships, employee:skill and employee:language, are independent in that there is no direct connection between skills and languages. There is only an indirect connection because they belong to some common employee (Kent 1983, 123).

This data independence implies that there is no restriction regarding pairing language with skill in a record since "the pairing does not convey any information" (Kent 1986, 123). This fact allows the three maintenance policies discussed for data records to exist.

> ... suppose that an employee can only exercise certain skills in certain languages. Perhaps Smith can cook French cuisine only, but can type French, German, and Greek. Then the pairing of skills and languages becomes meaningful, and there is no longer an ambiguity of maintenance policies. In the present case, only the following form is correct (see Table 4) (Kent 1983, 123).

Table 4. A Data Dependent Record.

| EMPLOYEE | SKILL | LANGUAGE |
|---|---|---|
| Smith | cook | French |
| Smith | type | French |
| Smith | type | German |

Source: Kent, W. "A Simple Guide To Five Normal Forms in Relational Database Theory." Communication of ACM 26, no. 2. (1986): 123.

> Thus, the employee:skill and employee:language relationships are no longer independent. These records do not violate fourth normal form. When there is an interdependence among the relationships, it is acceptable to represent them in a single record (Kent 1983, 123).

A fundamental problem is whether the further normalization from the third normal form to the higher normal form is needed.

> The dependencies are specified during the design of relation schemas and are used to decompose the initial schemas into smaller components. Whereas it is relatively easy to obtain an accurate set of FDs, it is difficult to determine a correct set of MVDs. MVDs depend on the context and thus are very hard to visualize ... (Jajodia 1986, 458).

That is, in order to achieve a fourth normal form, a deep understanding of data usage is needed.

## The Third and Fourth Normal Forms

Fundamental differences between the third normal form and the fourth normal form arise from several points of view. The focus of this discussion is on the logical difference that appears between these two normal forms. The database scheme used by Fagin illustrates the differences between these normalized forms and is presented below (1977, 268). Fagin pictures the unnormalized scheme as R=(CLASS, SECTION, STUDENT, MAJOR, EXAM, YEAR, INSTRUCTOR, RANK, SALARY, TEXT, DAY, ROOM). Functional dependencies that exist in the relation R are delineated in Figure 3. Furthermore, multivalued dependencies are depicted in Figure 4.

1) {CLASS, SECTION} -> INSTRUCTOR,
2) {CLASS, SECTION, DAY} -> ROOM,
3) STUDENT -> {MAJOR, YEAR},
4) INSTRUCTOR -> {RANK, SALARY}.

Figure 3. Functional Dependencies.
Source: Fagin, R. "Multivalued Dependencies and a New Normal Form for Relational Database." ACM TODS 2, no. 3. (September, 1977): 269.

5) {CLASS, SECTION} ->-> {STUDENT, MAJOR, EXAM, YEAR},
6) {CLASS, SECTION} ->-> {INSTRUCTOR, RANK, SALARY},
7) {CLASS, SECTION} ->-> TEXT,
8) {CLASS, SECTION} ->-> {DAY, ROOM}, ...
9) {CLASS, SECTION, STUDENT} ->-> EXAM

Figure 4. Multivalued Dependencies.
Source: Fagin, R. "Multivalued Dependencies and a New Normal Form for Relational Database." ACM TODS 2, no. 3. (September, 1977): 269.

In order to demonstrate the difference between the third and fourth normal forms, the original relation R must be decomposed into a set of smaller relations by using dependencies. The example is excerpted from Fagin (1977, 268-270) and will be presented at length below. By utilizing the four functional dependencies presented above in Figure 3, namely 1 through 4, the original relation R can be decomposed into a set of small third normal form relations {R1,R2,...R5}.

By using functional dependency #3, R can be decomposed into the following structure:

R1(STUDENT, MAJOR, YEAR) and
R2(CLASS, SECTION, STUDENT, EXAM, INSTRUCTOR, RANK, SALARY, TEXT, DAY, ROOM).

By using functional dependency #2, R1 can be decomposed into:

R2-1(CLASS, SECTION, DAY, ROOM) and
R2-2(CLASS, SECTION, STUDENT, EXAM, INSTRUCTOR, RANK, SALARY, TEXT).

By using functional dependency #4, R2-2 can be decomposed into:

R2-2-1(CLASS, SECTION, STUDENT, EXAM, INSTRUCTOR, TEXT) and
R2-2-2(INSTRUCTOR, RANK, SALARY).

By using functional dependency #1, R2-2-2 can be decomposed into:

R2-2-2-1(CLASS, SECTION, STUDENT, EXAM, TEXT ) and
R2-2-2-2(CLASS, SECTION, INSTRUCTOR).

As a result, we have the following third normal form generated as a data scheme:

1) R1(STUDENT, MAJOR, YEAR)
2) R2-1(CLASS, SECTION, DAY, ROOM)
3) R2-2-2(INSTRUCTOR, RANK, SALARY)
4) R2-2-2-1(CLASS, SECTION, STUDENT, EXAM, TEXT)
5) R2-2-2-2(CLASS, SECTION, INSTRUCTOR).

For convenience, call R2-1 R2, R2-2-2 R3, R2-2-2-1 R4, and R2-2-2-2 R5.

To decompose R into a set of fourth normal data relations the same rules are applied, with one addition.

Utilizing multivalued dependency #7, R2-2-2-1 (or R4) can be decomposed into:

R2-2-2-1-1(CLASS, SECTION, STUDENT, EXAM)
R2-2-2-1-2(CLASS, SECTION, TEXT).

For convenience, call R2-1 R2, R2-2-2 R3, R2-2-2-2 R4, R-2-2-2-1-1 R5, and R-2-2-2-1-2 R6. As a result, we have the following fourth normal form database schemes:

R1(CLASS, SECTION, MAJOR, YEAR)
R2(STUDENT, MAJOR, YEAR)
R3(INSTRUCTOR, RANK, SALARY)
R4(CLASS, SECTION, INSTURCTOR)
R5(CLASS, SECTION, STUDENT, EXAM)
R6(CLASS, SECTION, TEXT)

The difference between the third and fourth normal forms is clearly shown; normalization into the third normal form utilizes only functional dependency, whereas the fourth normal form normalization is free from restricted use of functional dependencies and multivalued

dependencies. Consequently the logical difference manifests itself as the database scheme which is a result of decomposition. In the third normal form scheme, relation R4(CLASS, SECTION, STUDENT, EXAM, YEAR) exists, while in the fourth normal form, R5(CLASS, SECTION, STUDENT, EXAM) and R6(CLASS, SECTION, TEXT) are present. By (natural) joining R5 and R6 in the fourth normal form scheme, the resulting relation is the same as R4 in the third normal form (see Figure 5).

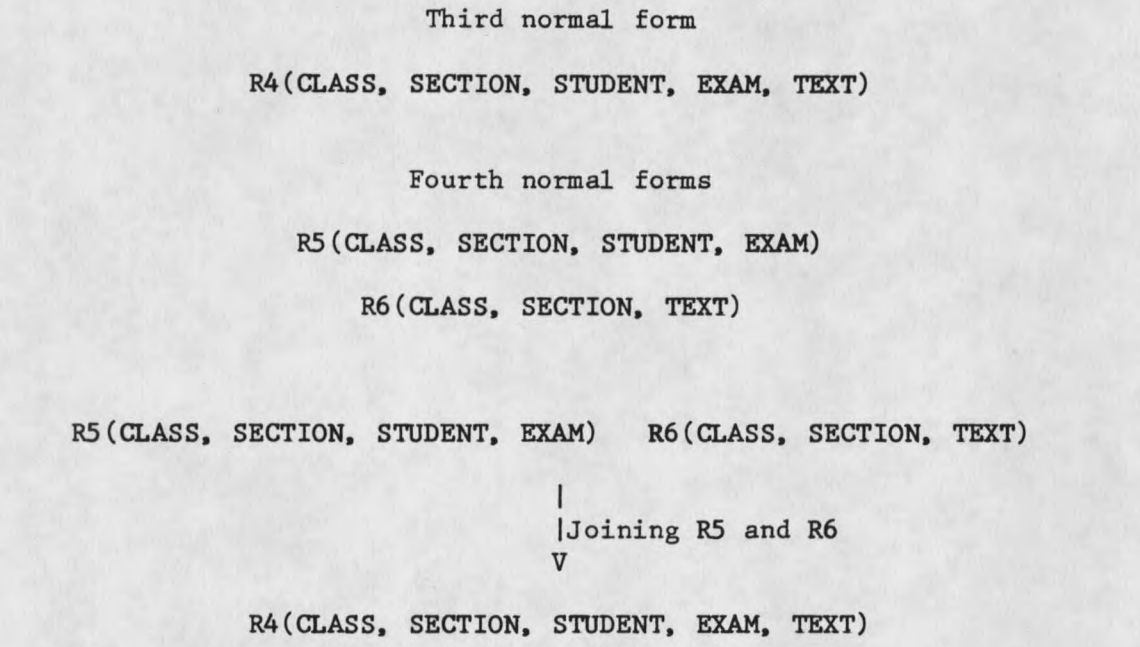



Figure 5. Lossless Decomposition From the Third Normal Form to the Fourth Normal Form.

In the third normal form scheme, for every occurrence of TEXT, the content of CLASS, SECTION, STUDENT, and EXAM must accompany the relation. All but one occurrence of CLASS, SECTION, STUDENT, and EXAM

are redundant. This data redundancy is described as follows: A relationship between EXAM and TEXT can be considered as a many-to-many relationship since every new occurrence of EXAM must be accompanied by a set of all occurrences of TEXT. If there are M number of EXAMs in the database and N number of TEXTs, the total number of tuples will be M*N. Another way of observing this redundancy is that all but one occurrence of CLASS, SECTION, STUDENT, and EXAM are redundant with respect to an occurrence of TEXT. During the reduction process from an unnormalized scheme to third normal form scheme, this type of redundancy is not observed. Table 5 clarifies this concept. In the second case in Table 5, where no multivalued dependency exists, the redundancy attributed to MAJOR is observed, but no other redundant implication is observed since a relationship between STUDENT and MAJOR is a one-to-many relationship. In the first case in Table 5, redundancy exists from EXAM to TEXT and TEXT to EXAM, and the relationship between EXAM and TEXT is a many-to-many projection. The relation R(CLASS, SECTION, STUDENT, EXAM, TEXT), a third normal form scheme, fails to meet the four criteria corresponding to normalization. Functional dependencies are used to create the third normal form scheme. The fourth normal form employs multivalued dependencies to improve the data scheme from the lower normal form.

One problem associated with the fourth normal form is that

> Whereas it is relatively to easy to obtain an accurate set of FDs, it is difficult to determine a correct set of MVDs. MVDs depend on the context and thus are very hard to visualize ... (Jajodia 1986, 458).

Table 5. Functional and Multivalued Dependencies

| CLASS | SECTION | STUDENT | EXAM | TEXT |
|---|---|---|---|---|
| C1 | S1 | ST1 | E1 | T1 |
| C1 | S1 | ST1 | E1 | T2 |
| C1 | S1 | ST1 | E2 | T1 |
| C1 | S1 | ST1 | E2 | T2 |

Multivalued dependency exists.

| CLASS | SECTION | STUDENT | MAJOR | EXAM |
|---|---|---|---|---|
| C1 | S1 | ST1 | M1 | E1 |
| C1 | S1 | ST1 | M1 | E2 |

Functional dependency exists.

The difference between the third and fourth normal forms can be summarized as follows: A database based on the third normal form is free from the four anomalies derived from functional dependency; however, some data redundancy is observed at this lower level of normalization. A fourth normal form data model can sustain a minimal data redundancy by utilizing multivalued dependencies. The main difficulty corresponding to the fourth normal form is to "determine a correct set of MVDs" (Jajodia 1986, 458). The motivation to measure efficiency between the third and the fourth normal forms is to discover which is more efficient with respect the considerations of space and time.

To summarize this chapter, when the original relation contains the multivalued dependency, there are two approaches to normalize the original relation to create a logical database scheme. One is to use

the third normal form and the other is to use the fourth normal form. If the third normal form is utilized, data redundancy exists. This data redundancy can be eliminated if the fourth normal form is used. The research presented in subsequent chapters was conducted to measure efficiency between these normalized schemes with respect to space and time.

# CHAPTER FIVE

## MATHEMATICAL VIEW OF STORAGE SPACE EFFICIENCY OF THE THIRD AND FOURTH NORMAL FORMS

### Introduction

This chapter gives induction proofs that the storage space required by a scheme in the third normal form grows multiplicatively as the complexity of data entities increases and that it grows additively for a scheme in the fourth normal form as complexity increases. The example database is based on Fagin (1977, 268-270). The cross product form in Chapter 4 is used to prove the storage size of the systems in the third normal form increase multiplicatively. This chapter focuses upon a section of the example database, in which a relation in the third normal form is different from a relation in the fourth normal form. The following relations based on the example provided by Fagin (1977, 268-270) and previously discussed will be used to support this argument:

Third Normal Form: R = {CLASS, SECTION, STUDENT, EXAM, TEXT}
Fourth Normal Form: R1 = {CLASS, SECTION, TEXT}
R2 = {CLASS, SECTION, STUDENT, EXAM}

## The Third Normal Form as a Multiplicative Model

In order to illustrate that the storage size of systems in the third normal form increase multiplicatively, a proof is derived by counting the number of tuples, as the number of TEXT and EXAM entities expand. Assume that the relation R(CLASS, SECTION, STUDENT, EXAM, TEXT) is composed of the following relational table:

| Tuple Number | CLASS | SECTION | STUDENT | EXAM | TEXT |
|---|---|---|---|---|---|
| 1 | C1 | S1 | ST1 | E1 | T1 |
| : | : | : | : | : | : |
| u | C1 | S1 | ST1 | En | Tm |

In order to visualize the entire database scheme, it is a best to use several different occurrences of CLASS, SECTION, and STUDENT; however, a proof requires that every occurrence of CLASS, SECTION, STUDENT becomes redundant as the number of EXAM and TEXT items increase.

### Basis Step

Basis: n = 1, m = 1.

When the number of EXAMs and the number of TEXTs is 1, the total number of tuples that exist in a database is the multiplication of the number of possible ways to extract a single data element of EXAMs from the total number, currently 1, by the number of possible ways to choose a single data element of TEXT from the total number of 1 TEXT.

$$\text{The number of tuples in the database} = \frac{1!}{1!\ 0!} * \frac{1!}{1!\ 0!} = 1 * 1 = 1$$

Induction Hypothesis

Induction hypothesis : m = M, and n = N.

If the number of EXAM occurrences is M, and the number of TEXT occurrences is N, then there will be M * N total number of tuples in the database (the third normal form) since the total number of tuples can be calculated as C1 * C2, where C1 represents the number of possible ways to choose an entity of EXAM from the total number of EXAMs M, while C2 represents the number of possible ways to choose an entity of TEXT from the total number of TEXTs N. That is, the total number of tuples in the database (third normal form) is:

$$\text{The number of tuples in the database} = \frac{M!}{1!(M-1)!} * \frac{N!}{1!(N-1)!} = M * N.$$

Therefore, the total number of tuples grows in a multiplicative fashion in the third normal form database model (see Table 7).

Table 7. Induction Hypothesis of the Third Normal Form Proof.

| Tuple Number | CLASS | SECTION | STUDENT | EXAM | TEXT |
|---|---|---|---|---|---|
| 1 | C1 | S1 | ST1 | E1 | T1 |
| : | : | : | : | : | : |
| M | C1 | S1 | ST1 | Em | T1 |
| M + 1 | C1 | S1 | ST1 | E1 | T2 |
| : | : | : | : | : | : |
| M + N | C1 | S1 | ST1 | Em | T2 |
| : | : | : | : | : | : |
| M * N | C1 | S1 | ST1 | Em | Tn |

Induction Step

Induction:

Assume that the induction hypothesis is true for all n and m, where $1 \leq m \leq M$, and $1 \leq n \leq N$. It must be shown that when $m = M + 1$ and $n = N$, the total number of tuples will be $(M+1)*N$. At the same time, the total number of tuples must be equal to the multiplication of the possible ways in which to choose one data element of EXAM with the total number of $N + 1$ and the possible ways to choose one data element of TEXT with the total number of M. The equality is exhibited by the following expressions:

$$(M+1)*N = \frac{(M+1)!}{M!\ 1!} * \frac{N!}{(N-1)!\ 1!}$$

$$\frac{(M+1)!}{M!\ 1!} * \frac{N!}{(N-1)!\ 1!} = \frac{M+1\ *\ M!}{M!\ 1!} * \frac{N\ *\ (N-1)!}{(N-1)!\ 1!} = (M+1)\ *\ N,$$

where M is the total number of EXAMs, and N is the total number of TEXTs.

Therefore, the proof indicates that the total number of tuples in the third normal form database grows in a multiplicative fashion (see Table 8).

By symmetry, as the number of TEXT entities increases by one, the total amount of space utilized by the third normal form is $M*(N+1)$. Therefore, the total number of tuples for a single occurrence of CLASS, SECTION, STUDENT grows in a multiplicative fashion.

Table 8. The Induction Step of the Third Normal Form Proof.

| Tuple Number | CLASS | SECTION | STUDENT | EXAM | TEXT |
|---|---|---|---|---|---|
| 1 | C1 | S1 | ST1 | E1 | T1 |
| : | : | : | : | : | : |
| M * N | C1 | S1 | ST1 | Em | Tn |
| M * N + 1 | C1 | S1 | ST1 | Em+1 | T1 |
| (M+1) * N | C1 | S1 | ST1 | Em+1 | Tn |

## The Fourth Normal Form as an Additive Model

The fourth normal form is additive with respect to the storage requirement. The mathematical proof is derived by counting the number of tuples as the number of TEXT and EXAM items increases. The relation R1(CLASS, SECTION, STUDENT, EXAM, TEXT) and R2(CLASS, SECTION, TEXT) appears as follows:

| Tuple Number | CLASS | SECTION | STUDENT | EXAM |
|---|---|---|---|---|
| 1 | C1 | S1 | ST1 | E1 |
| : | : | : | : | : |
| u1 | C1 | S1 | St1 | Em |

| Tuple Number | CLASS | SECTION | TEXT |
|---|---|---|---|
| 1 | C1 | S1 | T1 |
| : | : | : | : |
| u2 | C1 | S1 | Tn |

### Basis Step

Basis: n = 1, m = 1.

When the number of EXAMs and the number of TEXTs are 1, the total number of tuples that exist in a database is the addition of the number of possible ways to choose a single data element of EXAMs from the total number, currently 1, and the number of possible ways to choose a single data element of TEXTs from the total number of 1 TEXT entity.

$$\text{The tuples in the database} = \frac{1!}{1!\ 0!} + \frac{1!}{1!\ 0!} = 1 + 1 = 2$$

Therefore the total number of tuples in the database (fourth normal form) is 2 (see Table 9).

Table 9. Basis Step of the Fourth Normal Form Proof.

| Tuple Number | CLASS | SECTION | STUDENT | EXAM |
|---|---|---|---|---|
| 1 | C1 | S1 | ST1 | E1 |

| Tuple Number | CLASS | SECTION | TEXT |
|---|---|---|---|
| 1 | C1 | S1 | T1 |

Induction Hypothesis

Induction hypothesis : m = M, and n = N.

If the number of **EXAM** occurrences is M and the number of **TEXT** occurrences is N, then there will be M + N total number of tuples in the database (the fourth normal form) since the total number of tuples can be calculated as C1 + C2, where C1 represents the number of possible ways to choose an entity of **EXAM**s from the total number of **EXAM**s M, and C2 represents the number of possible ways to choose an entity of **TEXT**s from the total number of **TEXT**s N. That is, the total number of tuples in the database (fourth normal form) is as follows:

$$\text{The number of tuples in the database} = \frac{M!}{1!(M-1)!} + \frac{N!}{1!(N-1)!} = M + N.$$

Therefore, the total number of tuples grows in an additive fashion in the fourth normal form database model (see Table 10).

Table 10. Induction Hypothesis of the Fourth Normal Form Proof.

| Tuple Number | CLASS | SECTION | STUDENT | EXAM |
|---|---|---|---|---|
| 1 | C1 | S1 | ST1 | E1 |
| : | : | : | : | : |
| M | C1 | S1 | ST1 | Em |

| Tuple Number | CLASS | SECTION | TEXT |
|---|---|---|---|
| 1 | C1 | S1 | T1 |
| : | : | : | : |
| N | C1 | S1 | Tn |

Induction Step

Induction:

Assume that the induction hypothesis is true for all n and m, where $1 \leq m \leq M$, and $1 \leq n \leq N$. We must show that when $m = M + 1$ and $n = N$, the total number of tuples will be $(M+1)+N$. At the same time, the total number of tuples must be equal to the addition of the possible ways to choose a single data element of **EXAM**s with the total number of N + 1 and the possible ways to choose a single data element of **TEXT**s with the total number of M. The equality of this function is exhibited by the following equation:

$$(M+1)+N = \frac{(M+1)!}{M!\ 1!} + \frac{N!}{(N-1)!\ 1!}$$

$$\frac{(M+1)!}{M!\ 1!} + \frac{N!}{(N-1)!\ 1!} = \frac{M+1 * M!}{M!\ 1!} + \frac{N * (N-1)!}{(n-1)!\ 1!} = (M+1)+N,$$

where M is the total number of **EXAM**s, and N is the total number of **TEXT**s.

The proof, therefore, indicates that the total number of tuples in the fourth normal form database grows additively (See Table 11).

Table 11. Induction Step of the Fourth Normal Form Proof.

| Tuple Number | CLASS | SECTION | STUDENT | EXAM |
|---|---|---|---|---|
| 1 | C1 | S1 | ST1 | E1 |
| : | : | : | : | : |
| M | C1 | S1 | ST1 | Em |
| M + 1 | C1 | S1 | ST1 | Em+1 |

| Tuple Number | CLASS | SECTION | TEXT |
|---|---|---|---|
| 1 | C1 | S1 | T1 |
| : | : | : | : |
| N | C1 | S1 | TN |

By symmetry, as the number of TEXT entities increases by one, the total number of tuples is M+(N+1). Therefore, the total size of the tuple field for every single CLASS, SECTION, STUDENT will increase in additive fashion.

To summarize, a proof of the multiplicative growth of storage size in the third normal form scheme and a proof of the additive growth of storage size of the fourth normal form scheme are illustrated. These proof are important factors with regard to the efficiency measurement of both normalized form schemes. The theoretical proof provided here gives an indication of the importance of recognizing a multivalued dependency. As mentioned earlier in Chapter 4, a multivalued dependency seems difficult to detect; however, if the research hypothesis is correct, a substantial amount of memory will be saved when a multivalued dependency is present if further normalization into the fourth normal form is done.

# CHAPTER SIX

# MACHINE TIME EFFICIENCY MEASUREMENTS

## Introduction

Using machine comparisons, efficiency measurements between the third and the fourth normal forms were taken. A computer simulation method was developed to compare these two normal forms with respect to machine time required to execute a query. Such comparisons are necessary in order to test the time components of the hypothesis that the fourth normal form scheme requires less time to execute a query when compared with the third normal form scheme. In this chapter, the methodology for and results of the machine efficiency measurements are discussed. First, the need for the machine efficiency measurement approach is illustrated. Second, the methodology and the results of the machine efficiency measurements are discussed with respect to the machine space complexity. Finally, the methodology for and the results of the machine time efficiency measurements are presented.

## Need for the Machine Efficiency Measurement

In this section, several approaches to efficiency measurements are discussed. Teorey and Fry state two reasons for using efficiency measurements of the database to evaluate database schemes: 1) the

database must operate within "size constraints" and 2)The database must conform to the maximum allowable response time (1982, 33).

Atre agrees, and further insists that Teorey and Fry's two arguments support the idea of conducting the space (physical) and time performance tests as the efficiency measurements of a database scheme (1980, 288). From another point of view, Fedorowicz suggests that a database can be evaluated by measuring access time and response time. Fedorowicz's method involves a measurement of the individual user's interaction with the database, whereas other traditional methods involve "queuing models, simulation, and monitors of contrived or real workloads" (1987, 86). The user interaction approach will be explored in Chapter 7.

The importance of the size constraints and maximum allowable response time suggested by Teorey and Fry is illustrated by using the following scenario. When a user interacts with a database system, he/she must work under certain constraints. The constraints confronted by a user are categorized as the physical computer requirements and the time constraints regarding a query routine. The physical computer constraint can be described as a limitation of the actual computer's capacity. A user confronts the problems of physical input and output (I/O) constraints and memory size limitation whenever executing the database program.

The physical constraints are divided into three different components:

1) I/O
2) Central Processing Unit (CPU)
3) secondary storage

Though physical constraints are divided into three components, the main concern in regard to the efficiency measurement presented here is the third category, secondary storage. The actual physical I/O and CPU characteristics are relatively trivial with respect to this experiment since all measurements were conducted with the same computer and should therefore be constant. Furthermore, both the third and fourth schemes used the same program. Each normalized form scheme uses a different data dictionary, which resides in secondary storage. If any difference exists between the forms with respect to space, it should manifest itself in the amount of secondary storage spaces allocated and/or utilized. Given the premise that the storage space requirements of the third normal form grow multiplicatively and that the storage of the fourth normal form grow additively, then the third normal form scheme is more efficient at the initial stage. The fourth normal form scheme, however, become more efficient as the complexity of the data entities increases. The space efficiency measurement is the one of topics discussed in this chapter.

Another criterion of efficiency is the machine time required to execute a query. The time constraint measurements are divided into three different categories:

1) CPU time
2) I/O time
3) user's response time

CPU time and I/O time are combined and considered as the machine time in this thesis. The time measurement can be crucial in order to establish an efficiency difference between the third and fourth normal forms for the following two reasons:

1) A user must process the query within time constraints;
2) The time allocated to a query must be minimized for the database design to be appropriate;

Both of the above reasons can be explained in the following manner: if the computerized database requires more time to retrieve information than some other alternative method, then the computerized database is not needed. Since the user must endure the execution time, efficiency with regard to machine time is defined as the shortest execution time required to complete a user's query.

To summarize this section, two criteria of efficiency are discussed. One is the space allocated and used by the database. The other is the execution time taken by the database during the query. The needs for machine efficiency are twofold: one concerns the constraints which the user encounters, and the other is that appropriate design minimizes the space and time required by the database. A third efficiency criterion, user response time, is the subject of Chapter 7.

## Space Efficiency Measurement Methodologies

For the database to be efficient, the database should require the smallest amount of the computer memory possible. The more memory space required, the longer the database program needs to execute a relational operation. One object of this work is to detect the point where the fourth normal form scheme becomes more efficient than the third normal form scheme.

As the number of examinations (EXAM) and textbooks (TEXT) grows, the difference between the third and fourth normal forms should become more obvious (see Table 12). In the previous chapter, the mathematical proof was presented to show that the third normal form scheme increases its secondary space size multiplicatively, whereas the fourth normal form scheme increase its secondary memory size additively. The third normal form should be more efficient at the initial stage(s) of a query routine until the number of examinations (EXAM) and the number of textbooks (TEXT) both became greater than 2. Until that point the third normal form scheme requires less space (see Table 12).

Table 12. The Complexity of Data Relationship.

| Number of EXAM | Number of TEXT | Normal Forms Third | Fourth | Ratio |
|---|---|---|---|---|
| 1 | 1 | 1 | 2 | .50 |
| 1 | 2 | 2 | 3 | .66 |
| 1 | 3 | 3 | 4 | .75 |
| 1 | 4 | 4 | 5 | .80 |
| 1 | 5 | 5 | 6 | .83 |
| 2 | 1 | 2 | 3 | .66 |
| 2 | 2 | 4 | 4 | 1.0 |
| 2 | 3 | 6 | 5 | 1.2 |
| : | : | : | : | : |
| 5 | 5 | 25 | 10 | 2.5 |

Assume that both TEXT and EXAM take one unit of storage each.

In Table 12, the number in the 'Third' column is calculated by multiplying the number of examinations (EXAM) by the number of textbooks (TEXT). The number in the 'Fourth' column is calculated by adding the number of examinations (EXAM) to the number of textbooks (TEXT). The 'Ratio' column is the number in the 'Third' column divided by the number in the 'Fourth' column.

The actual measurement using the computer simulation was conducted by modifying the database program which was created by the author and used for the user's response time measurements for both the normalized forms. Modification was undertaken in the following manner. First, four query access paths were chosen by the author. These four query routines used were the same ones used by the end user in another part of these tests. Each query routine consisted of different combinations of data entities, and demonstrated that the access path for each normalized form requires a different access path. Second, the operations required for the query access paths were programmed into the database. Third, data files of different complexity were created, with the number of examinations (EXAM) ranging from 1 to 5; the number of textbooks (TEXT) also ranged from 1 to 5. Finally, the database program was executed 25 times for each query per normal form version. During the execution of the program, machine time was recorded.

Two methodologies were also developed to measure the space efficiencies according to the hypothesis that the fourth normal form scheme requires less space compared with the third normal form scheme during a query routine:

1) the ratio of bytes actually used to bytes actually allocated for both the third and the fourth normal forms;
2) the ratio of allocated space in the third normal form to allocated space in the fourth normal form, and the ratio of what is actually used in the third normal form to what is actually used in the fourth normal form.

The formula for the first ratio is a simple division of the actual space used by the physically allocated space for both the third normal form and the fourth normal form (the-actual-space-used/the-physically-allocated-space). This gives the relative space use of the third and the fourth normal form schemes.

Two formulas are used to calculate the second ratio. The first is a division of actual space used in the fourth normal form scheme by the actual space used in the third normal form scheme (actual-space-used-in-the-third-normal-form-scheme/actual-space-used-in-the-fourth-normal-form-scheme). The second is a division of the allocated space in the fourth normal form scheme by the allocated space in the third normal form scheme (allocated-space-in-the-third-normal-form/allocated-space-in-the-fourth-normal-form). These ratios show the relative usage and allocation of comparison of these two normalized forms schemes.

To summarize, the space efficiency measurement was conducted by calculating the efficient usage and the relative usage and allocation of the memory space by the two normalized form schemes for every query routine. The ratio form is used to make a relative comparison of the space efficiency in the normalized form schemes.

## Analysis of Space Efficiency Measurements

In this section, the results from the four query routines are discussed. In each query routine, three comparisons are made: the first comparison is between actually used space and physically allocated space. The second comparison is made by dividing the actual

space of the third normal form scheme by the actual space of the fourth normal form schemes. The third comparison is the ratio of physically allocated space in fourth normal form to physically allocated space in third normal form.

Analysis of The First Query Routine

During the first query routine, the program created a relation containing CLASS SECTION INSTRUCTOR STUDENT. The query access paths used by both the third and fourth normal form schemes are similar due to the nature of the first query (see Figure 9, Appendix A). The other reason to choose the similar access path is that the measurements are strictly based on the secondary storage utilized by both normalized forms. That is, other variables are constant. From the machine viewpoint, three distinct observations were made regarding query one.

One observation relates to the relative use of space by each normalized form scheme (see Table 60, Appendix B). The third normal form uses its space efficiently in terms of allocated space. This indicates that the third normal form scheme requires an allocation of less memory space than the fourth normal form scheme; however, the fourth normal form scheme requires less memory space to execute a query.

The second observation relates to the ratio of space actually used between the third and fourth normal form schemes. As the complexity of data entities increases, especially the number of examinations, the ratio of the actually utilized memory spaces between the third and fourth normal form schemes increases. When five

examinations and five textbooks are used for the data entities, the ratio becomes 5.56:1. That is, if the machine efficiency measurement is concerned only with the actual usage of storage size, the third normal form scheme needs to allocate 5.56 times more memory space than the fourth normal form scheme does.

The last observation is made with regard to the relative use of physically allocated space by each normalized form scheme. Even if the physically allocated space is measured, the third normal form scheme requires 3.74 times the memory space than does the fourth normal form.

In the efficiency measurement relating to memory space used during the first query routine, the fourth normal form scheme showed the best efficiency from even for simple data entities.

To summarize, in the actual space comparison between the third and fourth normal form schemes, the fourth normal form scheme is relatively more efficient. The third normal form scheme requires 1.14 times more memory space than the third normal form scheme requires to complete the first query with the complexity of data entities of one textbook and one examination. When the database is required to store five textbooks and five examinations, the third normal form scheme requires 5.56 times more memory space than the fourth normal form scheme to complete the same query routine. In allocated memory comparison, the third normal form scheme is relatively more efficient whenever the database is required to store only one examination.

Analysis of The Second Query Routine

The second query task is to create a relation containing INSTRUCTOR RANK STUDENT MAJOR. The query access paths utilized by both the third and the fourth normal forms are depicted in Figure 10, Appendix A. Both query access paths are similar. In regard to the machine efficiency data, two observations on space complexity need to be discussed. The trend of the ratio between the actually used space and the physically allocated space is the same as the first query. The ratios in both normalized form schemes, however, seem higher than the first query. The third normal form scheme used, at most, 80 per cent of its provided space, whereas the fourth normal form used, at most, 58 per cent (see Table 61, Appendix B). The higher ratio of space used during the second query routine implies that the execution time involved by the third normal form scheme must be very lengthy compared with the fourth normal form scheme. In Table 61, Appendix B, the ratio of actual memory space used illustrates that the fourth normal form required less space than the third normal form. The third normal form requires at worst 5.60 more memory space to execute the same query task. Even at the lower complexity of data entities, the fourth normal form scheme requires less memory space to complete the second query.

To summarize, the relative efficiency between the third and the fourth normal form schemes is illustrated during the execution of the second query. With respect to the ratio of actually used space and the ratio of allocated space comparisons, the results are very similar to results from the first query routine. During query two, the third

normal form scheme uses, at most, 80 per cent of its provided space while the fourth normal form scheme utilizes, at most, 58 per cent of allocated space.

Analysis of The Third Query Routine

The construction of a relation CLASS SECTION INSTRUCTOR TEXT was the object during the third query routine. This query routine illustrates that the normal form has a high impact on the number of commands needed to complete the query. The results indicates that from the machine view, a substantial amount of space and time will be saved if the fourth normal form is used. The query access path used by the fourth normal form required only one operation, whereas the third normal form required two operations (see Figure 11, Appendix A). The actual amount of space utilized by the third normal form model in regard to allocated space increases gradually, but the actual use of space by the fourth normal form model in regard to allocated space does not show any increase (see Figure 62, Appendix B). Accordingly, when the comparison between the actual space used by both normalized forms is measured, the fourth normal form scheme performs better than the third normal form scheme (see Figure 62, Appendix B). At one point in the query execution pathway the third normal form scheme requires 10.24 times more space than the fourth normal form scheme.

To summarize the observations from the third query routine, the fourth normal form scheme has shown a relative efficiency measurement over the third normal form scheme. In comparisons of physically allocated spaces by both the normalized form schemes, the overall

efficiency differences are relatively same as the other two query routines.

Analysis of The Fourth Query Routine

During the fourth query routine, the fourth normal form scheme proved its efficiency with respect to execution time. The fourth query task is to create a relation containing INSTRUCTOR EXAM MAJOR. The query access paths used by both normalized forms are illustrated in Figure 12, Appendix A. Both normalized forms used the same number of operations. The third normal form scheme used its allocated space at 74 per cent of maximum, whereas the fourth normal form scheme used 50 per cent of its allocated space at the maximum point of use of the provided space (see Table 63, Appendix B). The actual space used by both normalized form shows that the fourth normal form scheme required only 17 per cent of the space actually used by the third normal form scheme at one point.

To summarize the efficiency measurements from the simulation of execution of the four query routines, the fourth normal form scheme was shown to be relatively more efficient with respect to memory space allocated and actually used than the third normal form scheme. Even at the lower level of complexity of data entities, the fourth normal form scheme has been shown to be relatively more efficient than the third normal form scheme. If the actual usage of memory space is measured, the fourth normal form is relatively more efficient at any complexity of data entities. Even if the allocated memory comparison

is made, the detection of multivalued dependency will save a relatively large amount of space.

## Methodology for Time Efficiency Measurements

When the time aspect of database efficiency is measured, it is hard to achieve an objective analysis since there is no one correct way to create a desired relation. A trade-off occurs between the number of relational operations utilized and the time involved in processing the query, and there is more than one correct way to create a desired relation. To illustrate this trade-off, assume the following scenario: One has to create a relation with CLASS, SECTION, INSTURCTOR, STUDENT, MAJOR from the third normal form based database scheme illustrated in Figure 6.

R1(CLASS, SECTION, STUDENT, EXAM, TEXT)
R2(CLASS, SECTION, INSTRUCTOR)
R3(INSTRUCTOR, RANK, SALARY)
R4(STUDENT, MAJOR, YEAR)
R5(CLASS, SECTION, ROOM, DAY)

Figure 6. Third Normal Form Scheme.

One approach is to combine relation R1 and relation R2 and repeat the same procedure for combining the resulting relation with R3. Finally, project the resulting relation R7 to create a desired relation R8 (see Figure 7). The other approach is to project relation R1, and combine it with relation R2, and combine the resulting joined relation R7 with R4, then project the resulting relation R8 (see Figure 8).

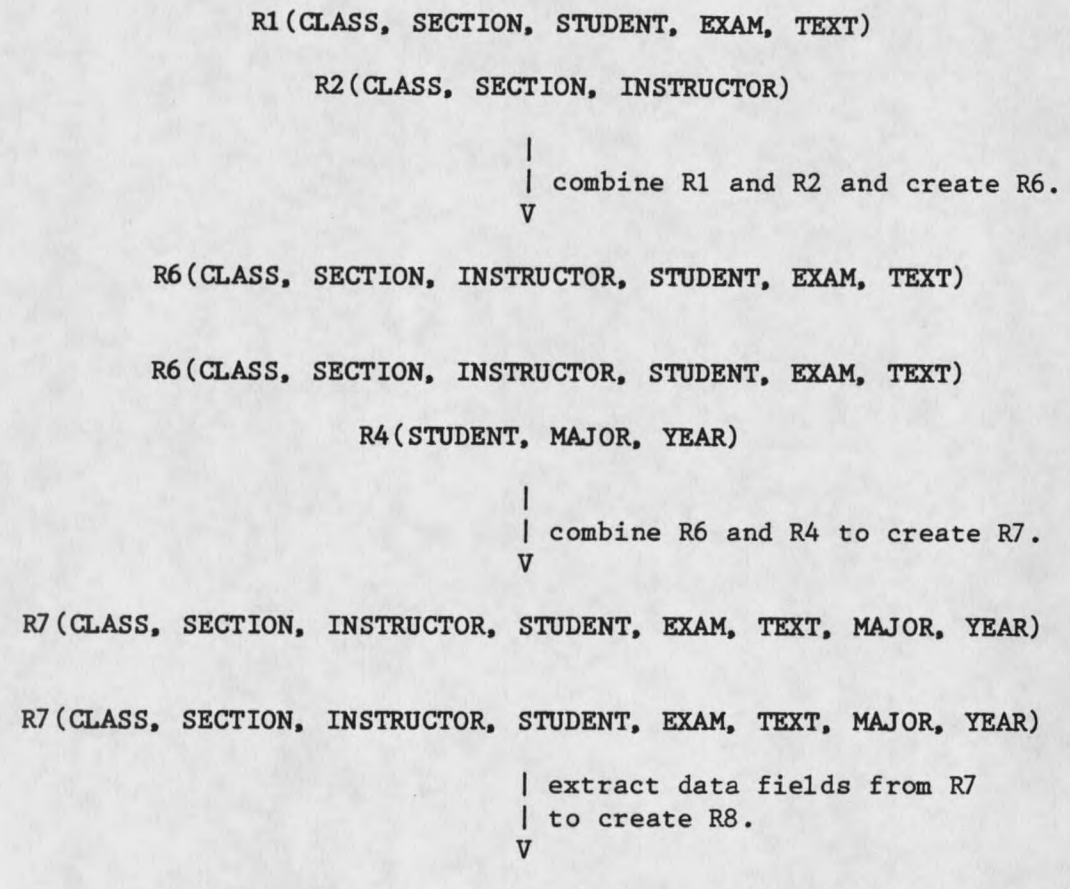



Figure 7. Approach One to a Query.

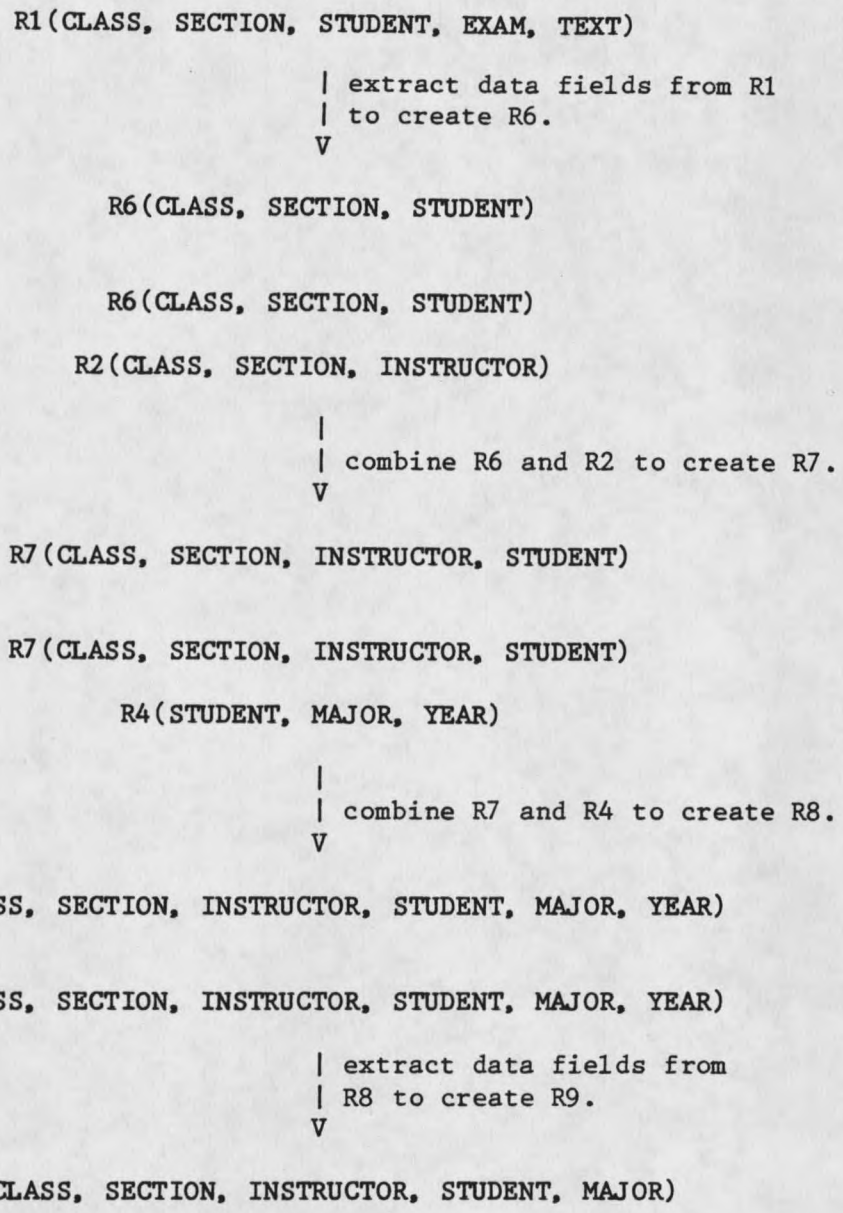



Figure 8. Approach Two to a Query.

The first approach requires three relational operations, whereas the second approach requires four relational operations. Table 13 illustrates the total time involved during the query process. The first approach takes more machine time than the second approach to

create a desired relation despite the fact that the second approach requires more relational operations. In general, if a data file consists of many data fields, it takes more machine time to join relations. This assumption is easily verified: when more data fields are used, more comparisons must be made in order to find the information to be joined into a new relation.

Table 13. Machine Time Comparison of Two Approaches to One Query Measured in units of 1/100 second.

| NUMBER OF EXAM | NUMBER OF TEXT | FIRST APPROACH | SECOND APPROACH |
|---|---|---|---|
| 1 | 1 | 4131 | 3876 |
| 1 | 2 | 4839 | 4132 |
| 1 | 3 | 5569 | 4356 |
| 2 | 1 | 4987 | 4256 |
| 2 | 2 | 5867 | 4387 |
| 2 | 3 | 6516 | 4503 |

The following example demonstrates why an objective analysis is difficult to achieve since there is no one correct way to create a desired relation. The pathway which will be chosen to find a particular relation depends on the user's perception of the database. This user's perception will be discussed later along with the final analysis of efficiency measurements with respect to the relative ease of use of the database.

Computer simulation experiments were done in a manner similar to the experiments of space complexity efficiency measurement. First, the query access paths for all four query routines were initially assigned using the same query paths as in the space complexity experiments. Second, the query access paths were programmed into the database program. Third, while the database and the query access

paths were unchanged, the database was executed with data entities of differing complexities. The machine (CPU and I/O) time required to execute each query was recorded. To measure time efficiency the ratio between the machine time used for the query routine in the third normal form and the machine time used for the query routine in the fourth normal form was found. The formula used to calculate the results presented in Appendix C is a division of the execution time for the third normal form scheme by the execution time for the fourth normal form scheme (the-execution-time-for-the-third-normal-form/the-execution-time-for-the-fourth-normal-form).

## Analysis of Time Efficiency Measurements

The results from the four query routines are discussed. In each query, the ratio between the machine time used for the query routine in the third normal form scheme and the machine time used for the query routine in the fourth normal form scheme are calculated.

### Analysis of the First Query Routine

The first query routine task is to create a relation containing CLASS SECTION INSTRUCTOR STUDENT. The query access paths of both the third and fourth normal form schemes were similar (see Figure 9, Appendix A). When the relationship between the number of examinations and the number of textbooks was 1:1, the third normal form scheme required 1.10 times the execution time that the fourth normal form

scheme required to complete the first query routine (see Table 64, Appendix C). When the relationship was 5:5, the third normal form scheme required 5.92 times the execution time that the fourth normal form scheme required to execute query one.

Analysis of the Second Query Routine

The second query routine task is to create a relation containing CLASS SECTION INSTRUCTOR RANK STUDENT MAJOR. Both the third and the fourth normal form schemes used similar access paths (see Figure 10, Appendix A). When the relationship between the number of examinations and the number of textbooks was 1:1, the third and fourth normal form schemes required almost the same amount of execution time to complete the second query (see Table 65, Appendix C). When the relationship was 5:5, the third normal scheme required 1.77 times the execution time to complete the second query that the fourth normal form scheme required.

Analysis of the Third Query Routine

For the third query routine, the desired relation is CLASS SECTION INSTRUCTOR TEXT (see Figure 11, Appendix A). When the relationship between the number of examinations and the number of textbooks was 1:1, it took the third normal form 1.97 times longer to execute than the fourth normal form (see Table 66, Appendix C). When the relationship was 5:5, the third normal form scheme required 18.21 times the machine retrieval time to complete the third query than the fourth normal form scheme needed for the same task.

Analysis of the Fourth Query Routine

The fourth query routine is to create the relation INSTRUCTOR EXAM MAJOR. The query access paths used by both the third and fourth normal forms are illustrated in Figure 12, Appendix A. The fourth query routine does not show an efficiency advantage for the fourth normal form scheme. When the relationship between examination and textbook was 1:1, it took the third normal form 1.04 times longer to execute than the fourth normal form. When the relationship was 5:5, the third normal form scheme required 1.13 times longer to execute than the fourth normal form scheme (see Table 67, Appendix C).

To summarize the efficiency measurement of time complexity with regard to machine time, the fourth normal form was more efficient in all trials. The degree of time efficiency depends on the query routine. During the third query routine, the fourth normal form scheme showed a relatively high efficiency compared with the third normal form scheme (18.21:1). The lowest efficiency advantages, observed during the fourth query routine, found the ratio of execution time of the third normal form to the fourth normal form to be, at most, 1.13:1. As the overall conclusion, the fourth normal form scheme is in general more efficient than the third normal form scheme, but the magnitude of the advantage of the fourth normal form over the third normal form depends upon the specific query and the degree of data complexity.

# CHAPTER SEVEN

# EFFICIENCY MEASUREMENTS OF USERS OF THE THIRD AND FOURTH NORMAL FORMS

## Introduction

In this chapter, the experiments to determine the response time of end users are discussed, including the methodologies used and the results obtained. Fedoriwicz suggested that the time factor in database efficiency measurements should include user response time as well as machine response time (1987, 86). When considering the issue of user response time, the factors which seemed to be important were training in computer use, general academic background, and cognitive style.

It was hypothesized that the important factors in user response time are general academic background and amount of computer training (major), and cognitive style. It was assumed that computer science and math majors would have the most knowledge of logical relationships and of how to use a computer efficiently and thus they would require the least time. On the other hand, business majors, who represent a typical end user of a database, can be expected to have less training in logic and computer science and would therefore take longer on the tasks measured. It was also assumed that the user's cognitive style would affect the time required to complete a query. The reason for considering cognitive style is that if the user's cognitive style is a

factor in determining the efficiency (speed or ease of use) of the database, the user's cognitive style must be considered during the database design.

## Methodology for User's Response Time Measurements

The measurements are based on the length of time it takes for a user to complete each of two query routines and the number of commands used by a user during each of the two routines.

The steps used during the experiment are as follows. Each user filled out a questionnaire (see Appendix D) prior to the experiment. Upon completion of the questionnaire, each user made a sample query together with the author after a brief explanation of the relational database and the relational commands. Each student used a Zenith personal computer to formulate and execute each of two queries. The normal form scheme used and the end goal of each query were provided by the author. The same four queries were used as in the simulation and were rotated among the students. Upon completion of the first query routine, the student was told to exit from the database system and wait for further instructions. If the student was not finished after 20 minutes of work, he/she was asked to go to the next task. Each student was instructed to make a second query into the database and exit from the data base when finished. During the experiment, the execution time and the number of commands utilized by the user were recorded on the diskette with which each student was provided. To conclude the experiment, the questionnaire and the diskette were collected.

To analyze the results, the data are described and a discussion of the statistical method is presented.

Data Description

The samples were collected from two computer science courses, two business courses, and a group of mathematics graduate students. During the analysis, the sample was divided into two groups: the computer science/mathematics group and the business group. The computer science/mathematics group refers to the computer professional group, whereas the business major group refers to a group of business users. The distribution of the user groups is depicted in Table 14. Table 14 illustrates the predominance of the business students at 61.09 per cent, whereas the computer science/mathematics majors comprise 38.64 per cent of the sample. Among the entire sample, males dominate the sample at 61.24 per cent (see Table 15).

Table 14. The Distribution of the User Group.

| | Size |
|---|---|
| Business major | 127 |
| Computer Science/ Mathematics | 81 |

Table 15. The Distribution of User's Sex.

| | Size |
|---|---|
| Male | 128 |
| Female | 80 |

Table 16 illustrates the distribution by the user's school year. Senior and graduate students dominate the sample at 48.57 per cent and indicate that the user's tended to be specialized with regard to academic major.

Furthermore, the number of students who used the third normal form scheme is 108, and the number of students who used the fourth normal form scheme is 102 (see Table 17). In order to further understand the sample group, several additional variables were used, although these variable were not used to analyze the efficiency measurements of any of the three approaches mentioned above (see Table 18).

Table 16. The Distribution of User's School Year.

| | Size |
|---|---|
| Freshman | 37 |
| Sophomore | 32 |
| Junior | 39 |
| Senior | 74 |
| Graduate | 28 |

Table 17. The Distribution of the Normalized Form Schemes.

| | Size |
|---|---|
| Third Normal Form Scheme | 108 |
| Fourth Normal Form Scheme | 102 |

Table 18. A List of Additional Variables for Descriptive Analysis.

| Variable | Description |
|---|---|
| Sex | Male or Female |
| Computer Experience | Weak, Average, or Strong background |
| Academic Experience | Weak, Average, or Strong background |

The user's computer experience refers to the length of time the user has been exposed to computers in general. A weak background is defined as being exposed to either a mainframe or personal computer for less than 17 months since on average a student has exposed to a computer for 17 months. The average background is defined as being exposed to the two kinds of computers for a combined time of more than or equal to 17 months but less than 33 months. Finally, the strong

background is defined as being exposed to microcomputers and mainframes for a total time of more than or equal to 33 months.

The user's academic experience refers to the combined number of computer related courses and mathematics courses taken by the user. A user with a weak background is defined as having had less than 17 credits of the combined computer related courses and mathematics courses since on average a student has 17 credits of the combined computer related courses and mathematics courses. A user with an average background is defined as one who has had more than and equal to 17 credits but less than 33 credits, and a user with a strong background is defined as one having more than equal to 33 credits of the combined computer related courses and mathematics courses.

The distribution of the users' academic experience, computer experience and academic major are discussed to justify the further analysis of data collected during the experiment. Table 19 illustrates the distribution of the user's academic experience versus the normalized form schemes. Table 19 shows that the distribution of users with respect to their academic background is evenly distributed.

Table 19. Academic Background Versus Normalized Forms.

| | Academic Background | | |
|---|---|---|---|
| | Weak | Average | Strong |
| Third Normal Form Scheme | 26 | 46 | 36 |
| Fourth Normal Form Scheme | 32 | 45 | 25 |

The distribution of users' computer experience versus the normalized form scheme is depicted in Table 20. Note that the sample is evenly distributed among the normalized form schemes.

The distribution of the business users and computer professional users versus the normalized form is illustrated in Table 21.

Table 20. Computer Background Versus Normalized Forms.

| | Computer Background | | |
|---|---|---|---|
| | Weak | Average | Strong |
| Third Normal Form Scheme | 36 | 45 | 27 |
| Fourth Normal Form Scheme | 40 | 41 | 21 |

Table 21. Normalized Forms Versus Users.

| | Business User | Computer Professional |
|---|---|---|
| Third Normal Form Scheme | 64 | 44 |
| Fourth Normal Form Scheme | 65 | 37 |

With respect to academic major, there are two distribution tables to illustrate the distribution within the sample. First, the distribution of user groups versus academic background is presented (see Table 22). Second, the distribution of user groups versus computer background is illustrated in Table 23.

Table 22. Users Versus Academic Background.

| | Academic Background | | |
|---|---|---|---|
| | Weak | Average | Strong |
| Computer Professional | 0 | 24 | 57 |
| Business User | 58 | 67 | 4 |

Table 23. Users Versus Computer Background.

| | Computer Background | | |
|---|---|---|---|
| | Weak | Average | Strong |
| Computer Professional | 7 | 37 | 37 |
| Business User | 69 | 49 | 11 |

During the analysis of the four query routines, four variables were used to analyze the efficiency measurements of both time and the number of relational operations required to complete a query and in the construction of a statistical model (see Table 24). The variable

Normal Form represents either the third or the fourth normal form scheme. The variable Academic Major, which is also referred to as user group, represents either a business major or a computer science/mathematics major. Cognitive Style 1 and Cognitive Style 2 represent the user's cognitive styles. Both cognitive styles are discussed in the methodology related section to cognitive style analysis.

Table 24. A List of Variables Used to Analyze User Response Time and the Number of Relational Operations.

| Variable | Explanation |
|---|---|
| Normal Form | Third or Fourth Normal Form |
| Major(user group) | Computer Science/Mathematics or Business |
| Cognitive Style 1 | Introverted or Extroverted |
| Cognitive Style 2 | Thinking or Feeling |

Data Analysis

The analysis of variance was the method used to analyze a user's response time. The analysis of variance is the methodology of choice, since the variables used to describe the user are nominal variables (Haber and Runyon 1969, 223-232).

Two dependent variables are used during the statistical analysis, one being the length of time involved during the query routine and the other being the number of relational operations used during the query routine. The length of time indicates the total of machine time and the time required by a user to analyze the access path and execute the query. The number of commands used by a user illustrates the relative ease of use corresponding to the database scheme. The smaller number

of commands indicates that the access path related to a specific query routine is easier and that a fewer number of mistakes may have been made by the user.

Several iterations of the analysis of variance were made in order to establish the appropriate statistical model. The statistical analysis of four different query routines was conducted using the SAS statistical package available at the Montana State University VAX 780 system. The analysis contains two components; the initial and detailed analyses are conducted under the general linear model procedure (analysis of variance method) (Ray 1982, 113-244).

The initial analysis evaluates all possible combinations of the variables and detects the important pairs. The detailed analyses select the most important variables to describe the statistical model based on the criterion of the p value being less than 0.05. After one or more iterations of the detailed analyses, the final model is presented. In the final model, the p value of the variable must be less than 0.05. The p value of 0.05 is chosen since

> When the event occurs five percent of the time or less, by chance, some researches are willing to assert that the results are due to nonchance factors (Haber and Runyon 1969, 173).

## Cognitive Style Methodology

An attempt was made to see if a user's cognitive style had an impact on the ease of use of the database system he/she used. Ease of use was measured by the number of relational operations and total time

required to do the query routines (including user response time and machine response time).

The scheme used to measure cognitive style is called the Myers-Briggs type indicator (Keirsey and Bates 1978, 13-26). There are two cognitive style relationships. One is called the "extroverted - introverted" relationship and the other is called the "thinking - feeling" relationship. The "extroverted - introverted" relationship can be summarized as follows:

> The main word which differentiates an extravert from an introvert is sociability as opposed to territoriality, but the extravert also finds breadth appealing where the introvert finds the notion of depth more attractive. Other notions which give a cue to this preference are the idea of external as opposed to internal; the extensive as opposed to the intensive; interaction as opposed to concentration; ... (Keirsey and Bates 1978, 16).

The "thinking-feeling" relationship is summarized as follows:

> Persons who prefer impersonal choice as a way of making decisions (T people) tend to response positively to such words as objective, ... while persons who choose in the context of the effects of the choice on themselves and others (F people) tend to react positively to words such as subjective, .... The T person tends to give priority to objective criteria, and is apt to be good at argumentation, attempting to win people over the his point of view through logic rather than appeal to the emotions. The F people tends to be good at persuasion and makes choices in the context of the personal impact of the decision on the people around him (Keirsey and Bates 1978, 22).

A questionnaire was developed by Ms. Sobel, the head of the Information Systems Department at Southern Methodist University of Dallas, Texas to determine the user's academic background, experience with computers, and cognitive style (see Appendix D). Based on answers from the questionnaire, the two styles of users' cognitive

styles were evaluated at Southern Methodist University. Ms. Sobel, evaluated the questionnaires and sent the cognitive style information for each user.

The research conducted attempted to show whether the user's cognitive style had an impact upon an efficiency measurement. If an impact is found, then a database designer must consider the user's cognitive style as another type of input to improve the database design.

## Analysis of the Four Query Routines

Previously the process was described to analyze the impact on the efficiency measurements of (1) the third and the fourth normal form schemes, (2) the business users and the computer professionals, and (3) the user's cognitive style. During the evaluation of each query routine, the overall statistical model is presented. After refining the initial statistical model, the detailed analysis is conducted to measure efficiency for each of the three aspects.

### Analysis of the First Query Routine

During the first query routine, the user tried to create a relation containing **CLASS SECTION INSTRUCTOR STUDENT.** Table 25 depicts the statistical model for the first query routine based upon the initial analysis. The query access paths used by both normalized forms are illustrated in Figure 9, Appendix A. Table 26 illustrates the refined analysis based on the results from the initial analysis of the first query routine.

The result based on the number of relational operations used during the first query does not show any statistical significance at p = 0.05 level. Thus, efficiency between the third and fourth normal data schemes is relatively identical. At p = 0.1 level, however, the difference between third and fourth normal form shows the statistical significance since the corresponding p value is 0.0591.

Table 25. The Initial Analysis of the First Query Routine Based on the Number of Relational Operations.

| | Degree of Freedom | F value | PR > F |
|---|---|---|---|
| Normal Form Schemes | 1 | 4.65 | 0.0345 |
| Academic Majors | 1 | 0.25 | 0.6169 |
| Cognitive Style 1 | 1 | 0.38 | 0.5407 |
| Cognitive Style 2 | 1 | 0.16 | 0.6882 |
| Normal Form Schemes X Academic Majors | 1 | 0.24 | 0.6282 |
| Normal Form Schemes X Cognitive Style 1 | 1 | 0.84 | 0.3624 |
| Normal Form Schemes X Cognitive Style 2 | 1 | 0.08 | 0.7763 |
| Normal Form Schemes X Cognitive Style 1 | 1 | 0.08 | 0.7763 |
| Academic Majors X Cognitive Style 2 | 1 | 1.70 | 0.1959 |

Table 26. The Detailed Analysis of the First Query Routine Based on the Number of Relational Operations.

| | Degree of Freedom | F value | PR > F |
|---|---|---|---|
| Normal Form Schemes | 1 | 3.66 | 0.0591 |
| Academic Majors | 1 | 0.00 | 0.9942 |
| Cognitive Style 1 | 1 | 0.00 | 0.9892 |
| Academic Majors X Cognitive Style 1 | 1 | 2.51 | 0.1170 |

The result from the computer simulation shows that the third normal form required 3.08 times the execution time to complete the first query than did the fourth normal form scheme (see Table 27). When the user's response time is added, the third normal form scheme

required 387.98 seconds to complete the first query exercise while the fourth normal form scheme needed only 291.59 seconds to complete the task, or a ratio of 1.33:1 (see Table 28).

Table 29 illustrates the statistical model based upon the execution time during the first query routine. Furthermore, Table 30 depicts the detailed analysis based on the first query routine.

Table 27. Machine Time Comparison Between the Normalized Forms During the First Query Routine.

| | Average Time (1 = 1 second) |
|---|---|
| Third Normal Form scheme | 201.85 |
| Fourth Normal Form scheme | 65.47 |
| Average | 133.66 |

Table 28. Time Comparison Between the Normalized Forms During the First Query Routine.

| | Sample Size | Average Time (1 = 1 second) |
|---|---|---|
| Third Normal Form Scheme | 45 | 387.98 |
| Fourth Normal Form Scheme | 41 | 291.59 |
| Computer Simulation Average | | 133.66 |

Table 29. The Initial Analysis of the First Query Based on the Execution Time.

| | Degree of Freedom | F value | PR > F |
|---|---|---|---|
| Normal Form Schemes | 1 | 9.05 | 0.0036 |
| Academic Majors | 1 | 0.04 | 0.8516 |
| Cognitive Style 1 | 1 | 0.12 | 0.7253 |
| Cognitive Style 2 | 1 | 0.04 | 0.8466 |
| Normal Form Schemes X Academic Majors | 1 | 0.26 | 0.6095 |
| Normal Form Schemes X Cognitive Style 1 | 1 | 12.51 | 0.0007 |
| Normal Form Schemes X Cognitive Style 2 | 1 | 2.97 | 0.0894 |
| Academic Majors X Cognitive Style 1 | 1 | 2.08 | 0.1535 |
| Academic Majors X Cognitive Style 2 | 1 | 3.48 | 0.0796 |

Table 30. The Detailed Analysis of the First Query Routine Based on the Execution Time.

| | Degree of Freedom | F value | PR > F |
|---|---|---|---|
| Normal Form Schemes | 1 | 9.35 | 0.0030 |
| Academic Majors | 1 | 0.64 | 0.4252 |
| Cognitive Style 1 | 1 | 1.09 | 0.3001 |
| Normal Form Schemes X Cognitive Style 1 | 1 | 14.25 | 0.0003 |

With respect to the time taken to complete the first query, the user's cognitive style refines the relative ease of use among the academic majors (see Table 31). Among the business majors, those who are classified cognitively as thinking types executed the query faster than those who are classified as feeling types; the ratio of thinking to feeling types is 1.48:1. Among computer science/mathematics majors, those who are classified cognitively as thinking retrieved the first query goal faster than those who are classified as feeling; the ratio of thinking to feeling types is 1.12:1. When the comparisons between academic majors and cognitive style 1 were independently analyzed, there exists no statistical significance (see Table 30). This finding implies that the user's cognitive style has more impact upon computer science/mathematics majors than upon business majors. Thus, an individual who tends to like analysis and putting things into logical order requires more time to complete a query.

To summarize, during the first query, measurements based on the user's response time show statistically significant difference with regard to normal form, cognitive style, and academic major. There is an indication of the importance of the user's cognitive style.

Table 31. Time Comparison Between Academic Major and Cognitive Style 2.

| | Sample Size | Average Time (1 = 1 second) |
|---|---|---|
| Business Major & Thinking | 18 | 459.79 |
| Business Major & Feeling | 27 | 310.72 |
| Computer Science/ Mathematics & Thinking | 20 | 322.03 |
| Computer Science/ Mathematics & Feeling | 22 | 304.38 |
| Average | | 342.58 |
| Computer Simulation Average | | 133.66 |

Analysis of the Second Query Routine

During query two, the user tried to create a relation containing INSTRUCTOR RANK STUDENT MAJOR. The query access paths utilized by both third and fourth normal forms are depicted in Figure 10, Appendix A. Both queries seem to use similar access path as shown in Figure 10, Appendix A. The statistical results from the second query routine illustrate some interesting results (see Tables 32 and 33).

The number of relational operations differs between business majors and computer science/mathematics majors. Generally, each operation adds to the time required to execute a query. Thus, a difference has occurred between the computer professionals and the business users. Table 34 illustrates the fact that the ratio between the business majors and computer science/mathematics majors is 1.49. That is, computer science/mathematics majors tended to require less time (about 50 percent less) than that required by business majors to complete query two. This fact is consistent with the previous results from the analysis of query one.

Table 32. The Initial Analysis of the Second Query Routine Based on the Number of Relational Operations.

| | Degree of Freedom | F value | PR > F |
|---|---|---|---|
| Normal Form Schemes | 1 | 2.01 | 0.1608 |
| Academic Majors | 1 | 20.81 | 0.0001 |
| Cognitive Style 1 | 1 | 0.49 | 0.4864 |
| Cognitive Style 2 | 1 | 0.11 | 0.7418 |
| Normal Form Schemes X Academic Majors | 1 | 0.52 | 0.4722 |
| Normal Form Schemes X Cognitive Style 1 | 1 | 0.00 | 0.9536 |
| Normal Form Schemes X Cognitive Style 2 | 1 | 0.12 | 0.7342 |
| Academic Majors X Cognitive Style 1 | 1 | 0.04 | 0.8498 |
| Academic Majors X Cognitive Style 2 | 1 | 4.31 | 0.0558 |

Table 33. The Detailed Analysis of the Second Query Routine Based on the Number of Relational Operations.

| | Degree of Freedom | F value | PR > F |
|---|---|---|---|
| Academic Majors | 1 | 13.12 | 0.005 |

Table 34. The Comparison of the Number of Relational Operations Between Academic Majors During the Second Query Routine.

| | Sample Size | Average Number of Operations |
|---|---|---|
| Computer Science/ Mathematics Majors | 28 | 5.82 |
| Business Majors | 56 | 8.67 |
| Average | | 7.72 |
| Computer Simulation Average | | 4.00 |

Table 35. Machine Time Comparison Between the Normalized Forms During the Second Query Routine.

| | Average Time (1 = 1 second) |
|---|---|
| Third Normal Form scheme | 307.98 |
| Fourth Normal Form scheme | 234.02 |
| Average | 271.00 |

Table 35 illustrates the time comparison between the third and fourth normal form schemes during the computer simulation. Compared with the first query routine, the difference is relatively small. During the first query routine, the ratio of the machine time between the third and fourth normal form data scheme is 3.08:1 while the ratio of user response times associated with the third and the fourth normal models during the second query routine is 1.31:1. The statistical model based upon the execution time is depicted in Table 36. Table 37 depicts the detailed analysis of the second query routine based on the machine time to complete query two.

Table 36. The Initial Analysis of the Second Query Routine Based on the Execution Time.

| | Degree of Freedom | F value | PR > F |
|---|---|---|---|
| Normal Form Schemes | 1 | 3.28 | 0.0745 |
| Academic Majors | 1 | 21.00 | 0.0001 |
| Cognitive Style 1 | 1 | 0.75 | 0.3893 |
| Cognitive Style 2 | 1 | 1.84 | 0.1798 |
| Normal Form Schemes X Academic Majors | 1 | 0.42 | 0.5200 |
| Normal Form Schemes X Cognitive Style 1 | 1 | 3.27 | 0.0744 |
| Normal Form Schemes X Cognitive Style 2 | 1 | 5.29 | 0.0245 |
| Academic Majors X Cognitive Style 1 | 1 | 0.00 | 0.9970 |
| Academic Majors X Cognitive Style 2 | 1 | 1.12 | 0.2931 |

Table 37. The Detailed Analysis of the Second Query Routine Based on the Execution Time.

| | Degree of Freedom | F value | PR > F |
|---|---|---|---|
| Normal Form Schemes | 1 | 7.01 | 0.0098 |
| Academic Majors | 1 | 23.48 | 0.0001 |

Table 38. Time Comparison Between the Normalized Forms During the Second Query Routine.

| | Sample Size | Average Time (1 = 1 second) |
|---|---|---|
| Third Normal Form Scheme | 34 | 773.35 |
| Fourth Normal Form Scheme | 50 | 672.49 |
| Average | | 713.32 |
| Computer Simulation Average | | 271.00 |

The fourth normal form model shows a shorter execution time (see Table 38) regarding query execution. The ratio between the third and the fourth normal form schemes is 1.15:1. The impact of the academic major upon time execution for query two is illustrated in Table 39. The business students took 1.6 times longer to complete the second query than did computer science/mathematics majors. This result is consistent with the overall efficiency measurement of the database as presented for the first query routine.

Table 39. Time Comparison Between Academic Majors During the Second Query Routine.

| | Sample Size | Average Time (1 = 1 second) |
|---|---|---|
| Business Major | 28 | 822.93 |
| Computer Science/ Mathematics Major | 56 | 494.10 |
| Average | | 713.32 |
| Computer Simulation Average | | 271.00 |

To summarize, with regard to the number of relational operations used during query two, no evidence was found that the fourth normal form data scheme is relatively easier for a user making query two into the database. Nevertheless, with respect to the execution time, evidence of relative ease of use by the fourth normal database scheme is observed based on shorter response times. Another observation is

that the differences between two academic majors manifests itself as the indication that prior computer and logic training affects the relative ease of use of the database.

Analysis of the Third Query Routine

The construction of a relation CLASS SECTION INSTRUCTOR TEXT is conducted during the third query routine. The query access paths used by both normalized forms are illustrated in Figure 11, Appendix A. Results from the third query routine illustrate the impact of the normal form scheme on completion time and the number of commands needed to complete the query. The number of relational operations required by the third normal form scheme was 2.5 times more than the number of relational operations required by the fourth normal form scheme during the execution of query three (see Table 40). Table 41 illustrates the statistical model based upon the number of relational operations used during the third query routine. Table 42 illustrates the detailed analysis of the third query routine. Table 43 illustrates the significant difference among the academic majors.

Table 40. The Comparison of the Number of Relational Operations Between the Normalized Forms During the Third Query Routine.

| | Sample Size | Average Number of Operations |
|---|---|---|
| Third Normal Form Scheme | 53 | 4.09 |
| Fourth Normal Form Scheme | 57 | 1.64 |
| Average | | 2.82 |
| Computer Simulation Average | | 1.50 |

Table 41. The Initial Analysis of the Third Query Routine Based on the Number of Relational Operations.

| | Degree of Freedom | F value | PR > F |
|---|---|---|---|
| Normal Form Schemes | 1 | 14.81 | 0.0002 |
| Academic Majors | 1 | 2.72 | 0.1027 |
| Cognitive Style 1 | 1 | 1.49 | 0.2254 |
| Cognitive Style 2 | 1 | 0.24 | 0.6221 |
| Normal Form Schemes X Academic Majors | 1 | 4.27 | 0.0416 |
| Normal Form Schemes X Cognitive Style 1 | 1 | 1.14 | 0.2881 |
| Normal Form Schemes X Cognitive Style 2 | 1 | 0.79 | 0.3767 |
| Academic Major X Cognitive Style 1 | 1 | 4.59 | 0.0357 |
| Academic Majors X Cognitive Style 2 | 1 | 1.96 | 0.1664 |

Table 42. The Detailed Analysis of the Third Query Routine Based on the Number of Relational Operations.

| | Degree of Freedom | F value | PR > F |
|---|---|---|---|
| Normal Form Schemes | 1 | 20.93 | 0.0001 |
| Academic Major | 1 | 1.69 | 0.1960 |
| Normal Form Schemes X Academic Major | 1 | 5.20 | 0.0246 |

Table 43. The Comparison of the Number of Relational Operations Between Academic Majors and the Normalized Form.

| | Sample Size | Average Number of Operations |
|---|---|---|
| Third Normal Form Scheme X Business Majors | 64 | 4.44 |
| Third Normal Form Scheme X Computer Science/ Mathematics Majors | 44 | 1.94 |
| Fourth Normal Form Scheme X Business Majors | 65 | 2.92 |
| Fourth Normal Form Scheme X Computer Science/ Mathematics Majors | 37 | 1.52 |

Business students demonstrated the most impact from the choice of normalized database scheme. Business students who used the third normal form data scheme required an average of 4.44 operations to complete query three, whereas business students who used the fourth normal form needed 1.94 commands. While the computer professionals who used the third normal form data scheme required 2.91 commands, the computer professionals using the fourth normal form needed 1.52 commands to complete a query. The differences between the third and fourth normal forms illustrate that users of the third normal form database scheme required approximately twice as many relational operations to construct the desired relation, regardless of background.

The computer simulation illustrates that the third normal form required 2.88 times more machine time to execute the query in third form than the higher normalization model (see Table 44). The statistical model used for the execution of query three based upon the execution time illustrates one statistical finding (see Table 46).

Table 44. Machine Time Comparison Between the Normalized Forms During the Third Query Routine.

| | Average Time (1 = 1 second) |
|---|---|
| Third Normal Form Scheme | 122.16 |
| Fourth Normal Form Scheme | 42.42 |
| Average | 82.22 |

Table 45. The Initial Analysis of the Third Query Routine Based on the Execution Time.

| | Degree of Freedom | F value | PR > F |
|---|---|---|---|
| Normal Form Schemes | 1 | 22.77 | 0.0001 |
| Academic Majors | 1 | 1.99 | 0.1616 |
| Cognitive Style 1 | 1 | 0.01 | 0.9261 |
| Cognitive Style 2 | 1 | 1.02 | 031631 |
| Normal Form Schemes X Academic Majors | 1 | 2.67 | 0.1057 |
| Normal Form Schemes X Cognitive Style 1 | 1 | 0.16 | 0.6872 |
| Normal Form Schemes X Cognitive Style 2 | 1 | 1.46 | 0.2297 |
| Academic Majors X Cognitive Style 1 | 1 | 0.34 | 0.5586 |
| Academic Majors X Cognitive Style 2 | 1 | 0.07 | 0.7911 |

Table 46. The Detailed Analysis of the Third Query Routine Based on the Execution Time.

| | Degree of Freedom | F value | PR > F |
|---|---|---|---|
| Normal Form Schemes | 1 | 50.62 | 0.0001 |

The initial analysis based on the execution time to complete query three is illustrated in Table 45. The user who utilized the third normal form required 2.4 times more execution time than that required to complete the third query (see Table 47).

Table 47. Time Comparison Between the Normalized Forms During the Third Query Routine.

| | Sample Size | Average Time (1 = 1 second) |
|---|---|---|
| Third Normal Form Scheme | 53 | 582.90 |
| Fourth Normal Form Scheme | 57 | 242.31 |
| Average | | 406.41 |
| Computer Simulation Average | | 82.22 |

To summarize, with respect to the number of relational operations, there are two findings. One is that there is a statistical significance between the two normalized forms. The other is that when a statistical significance between two normalized forms is analyzed along with academic majors, the fourth normal form scheme is relatively easier to use, especially by business users. The execution time comparison also supports the argument of the relative ease of use of the fourth normal form scheme.

Analysis of the Fourth Query Routine

During the fourth query routine, the fourth normal form scheme again proved its efficiency with respect to execution time and to the number of operations required to complete the query task. The construction of a relation INSTRUCTOR EXAM MAJOR is conducted during the fourth query routine. The query access paths used by both normalized forms are illustrated in Figure 12, Appendix A.

The statistical model for the execution time during the fourth query is illustrated in Table 48. Table 49 illustrates the detailed analysis of query four. An analysis from the user perspective indicates that users of the third normal form scheme required an average of 6.80 commands to execute this query routine while those who used the fourth normal form scheme required 4.83 commands to complete the query routine (see Table 50).

Table 48. The Initial Analysis of the Fourth Query Routine Based on the Number of Relational Operations.

| | Degree of Freedom | F value | PR > F |
|---|---|---|---|
| Normal Form Schemes | 1 | 7.31 | 0.0079 |
| Academic Majors | 1 | 15.63 | 0.0001 |
| Cognitive Style 1 | 1 | 3.97 | 0.0488 |
| Cognitive Style 2 | 1 | 3.30 | 0.0719 |
| Normal Form Schemes X Academic Majors | 1 | 6.06 | 0.0153 |
| Normal Form Schemes X Cognitive Style 1 | 1 | 0.27 | 0.6051 |
| Normal Form Schemes X Cognitive Style 2 | 1 | 0.86 | 0.3567 |
| Academic Majors X Cognitive Style 1 | 1 | 1.09 | 0.4194 |
| Academic Majors X Cognitive Style 2 | 1 | 0.66 | 0.4194 |

Table 49. The Detailed Analysis of the Fourth Query Routine Based on the Number of the Relational Operations.

| | Degree of Freedom | F value | PR > F |
|---|---|---|---|
| Normal Form Schemes | 1 | 7.33 | 0.0077 |
| Academic Majors | 1 | 15.02 | 0.0002 |
| Cognitive Style 1 | 1 | 5.18 | 0.0245 |
| Normal Form Schemes X Academic Majors | 1 | 6.47 | 0.0122 |

Table 50. The Comparison of the Number of Relational Operations Between the Normalized Forms During the Fourth Query Routine.

| | Sample Size | Average Number of Operations |
|---|---|---|
| Third Normal Form Scheme | 77 | 6.80 |
| Fourth Normal Form Scheme | 54 | 4.83 |
| Average | | 5.99 |
| Computer Simulation | | 3.50 |

The number of relational operations used during the query process differs between the academic majors. Business students required 1.73 times more relational operations to complete the fourth query when compared with computer science/mathematics majors (see Table 51).

Table 51. The Comparison of the Number of Relational Operations Between the Normalized Forms During the Fourth Query Routine.

| | Sample Size | Average Number of Operations |
|---|---|---|
| Business Major | 60 | 7.43 |
| Computer Science/ Mathematics Major | 71 | 4.28 |
| Average | | 5.99 |
| Computer Simulation | | 3.55 |

Cognitive style is shown to be a factor in the relative ease of use of the database. Table 52 shows evidence that those who tended to dislike complicated procedures (cognitive style 1 classification) required more relational operations to complete query four than did other students. When the user's cognitive style was analyzed with regard to academic major, great differences were found (see Table 53). Among the business students, those who are classified as extroverted types used 8.57 commands during the fourth query routine while those who are classified as introverted types used only 5.46 commands. The same pattern is exhibited with computer science/mathematics majors, although to a much lesser degree. It is presumed that the computer professionals' prior training reduced those differences which naturally result from one's cognitive style.

Table 52. The Comparison of the Number of Relational Operations Between the User's Cognitive Style 1 Classification During the Fourth Query Routine.

| | Sample Size | Average Number of Operations |
|---|---|---|
| Extroverted | 74 | 6.70 |
| Introverted | 57 | 5.07 |

Table 53. The Comparison of the Number of Relational Operations Between Academic Majors and the User's Cognitive Style 1 Classification During the Fourth Query Routine.

| | Sample Size | Average Number of Operations |
|---|---|---|
| Business Major X Extroverted | 45 | 8.57 |
| Business Major X introverted | 26 | 5.46 |
| Computer Science/ Mathematics X Extroverted | 32 | 4.31 |
| Computer Science/ Mathematics & Introverted | 28 | 4.25 |
| Average | | 5.99 |
| Computer Simulation Average | | 3.55 |

The computer simulation produced the machine time ratio of 1:2 (see Table 54). That is, the third normal form scheme tends to require twice as much time to execute query four as for the fourth normal form. Table 55 illustrates the initial analysis of query four. The statistical model has shown three findings with respect to the execution time of query four (see Table 56). The third normal form scheme required 1.43 more time than that required by the fourth normal form to complete the fourth query routine (see Table 57).

Table 54. Machine Time Comparison Between the Normalized Forms During the Fourth Query Routine.

| | Average Time (1 = 1 second) |
|---|---|
| Third Normal Form scheme | 219.12 |
| Fourth Normal Form scheme | 100.01 |
| Average | 159.65 |

Table 55. The Initial Analysis of the Fourth Query Routine Based on the Execution Time.

| | Degree of Freedom | F value | PR > F |
|---|---|---|---|
| Normal Form Schemes | 1 | 11.32 | 0.0010 |
| Academic Majors | 1 | 8.72 | 0.0038 |
| Cognitive Style 1 | 1 | 3.13 | 0.0797 |
| Cognitive Style 2 | 1 | 0.11 | 0.7452 |
| Normal Form Schemes X Academic Majors | 1 | 4.03 | 0.0471 |
| Normal Form Schemes X Cognitive Style 1 | 1 | 0.00 | 0.9580 |
| Normal Form Schemes X Cognitive Style 2 | 1 | 0.20 | 0.6520 |
| Academic Majors X Cognitive Style 1 | 1 | 1.16 | 0.2845 |
| Academic Majors X Cognitive Style 2 | 1 | 0.13 | 0.7151 |

Table 56. The Detailed Analysis of the Fourth Query Routine Based on the Execution Time.

| | Degree of Freedom | F value | PR > F |
|---|---|---|---|
| Normal Form Schemes | 1 | 11.39 | 0.0010 |
| Academic Majors | 1 | 12.89 | 0.0005 |
| Normal Form Schemes X Academic Majors | 1 | 5.74 | 0.0181 |

Another comparison is based upon academic major. The business students required 799.96 seconds to complete the fourth query routine while computer science/mathematics majors required 542.63 seconds during the fourth query routine (see Table 58). The final analysis regarding query four is based on the interaction of the normalized form data scheme with the different groups. Table 59 indicates that the normalized form scheme has more impact upon the business users than upon the computer scientists, in the sense that it takes the business user longer. This finding is consistent with the results from the other query routines.

Table 57. Time Comparison Between the Normalized Forms During the Fourth Query Routine.

| | Sample Size | Average Time (1 = 1 second) |
|---|---|---|
| Third Normal Form Scheme | 77 | 777.72 |
| Fourth Normal Form Scheme | 54 | 545.75 |
| Average | | 682.10 |
| Computer Simulation Average | | 159.65 |

Table 58. Time Comparison Between Academic Majors During the Fourth Query Routine.

| | Sample Size | Average Time (1 = 1 second) |
|---|---|---|
| Business Major | 60 | 799.96 |
| Computer Science/ Mathematics Major | 71 | 542.63 |
| Average | | 682.10 |
| Computer Simulation Average | | 159.65 |

Table 59. Time Comparison Between Academic Majors and the Normalized Forms During the Fourth Query Routine.

| | Sample Size | Average Time (1 = 1 second) |
|---|---|---|
| Business Major X Third Normal Form Scheme | 45 | 925.56 |
| Business Major X Fourth Normal Form Scheme | 26 | 582.57 |
| Computer Science/ Mathematics X Third Normal Form Scheme | 32 | 569.82 |
| Computer Science/ Mathematics X Fourth Normal Form Scheme | 26 | 511.57 |

During the first and second query routines, the query access paths utilized by both the normalized forms seem similar. No statistical significance is found with respect to the number of relational operations since the corresponding p value is greater than 0.05. When the query access paths utilized by both the normalized forms differ from each other, the fourth normal form scheme found to be easier for a user to use compared with the third normal form

scheme. This finding is possibly due to the logic to complete a query routine. Results from the third and fourth query routines support this argument by illustrating the impact of the user's cognitive style and the user's academic major on the results.

During the third query routine, business students demonstrated the most impact from the choice of normalized database scheme. Business students who used the third normal form data scheme required an average of 4.44 operations to complete query three, whereas business students who used the fourth normal form needed 1.94 commands. While the computer professionals who used the third normal form data scheme required 2.91 commands, the computer professionals using the fourth normal form needed 1.52 commands to complete a query. The differences between the third and fourth normal form illustrate that the user of the third normal form database scheme required approximately twice as many relational operations to construct the desired relation.

During the fourth query routine, a similar result is also found. Among the business students, those who are classified as extroverted types used 8.57 commands during the fourth query routine while those who are classified as introverted types used only 5.46 commands. The same pattern is exhibited with computer science/mathematics majors, although to a much lesser degree. It is presumed that the computer professionals' prior training reduced those differences which naturally result from one's cognitive style. Thus, when the query access paths seem similar for both the normalized forms, a statistically significant difference is not found. When the query

access paths seem to differ from each other, a combination of the user's cognitive style, the amount of the prior computer training, and the choice of normalized database scheme produce the statistically significant results.

With respect to the execution time, which includes both the actual computation time and user's response time, all four query routines illustrate the statistically significant difference between the normalized forms.

When the user's response time is measured during the first query routine, the third normal form scheme required 387.98 seconds to complete the first query exercise while the fourth normal form scheme needed only 291.59 seconds to complete the task, or a ratio of 1.33:1. Among the business majors, those who are classified cognitively as thinking types executed the query faster than those who are classified as feeling types; the ratio of thinking to feeling types is 1.48:1. Among computer science/mathematics majors, those who are classified cognitively as thinking retrieved the first query goal faster than those who are classified as feeling; the ratio of thinking to feeling types is 1.12:1. This finding implies that the user's cognitive style has more impact on business majors than on computer science/mathematics majors. Similarly, during the second query routine, the fourth normal form scheme shows a shorter execution time regarding query execution. The ratio between the third and the fourth normal form schemes is 1.15:1 and the the ratio between business students and computer science/mathematics majors is 1.60:1 during the second query routine.

To summarize results from the third and fourth query routine, the user who utilized the third normal form required 2.4 time more execution time than that required to complete the third query. The third normal form scheme required 1.43 time more time than that required by the fourth normal form to complete the fourth query routine. The ratio between the business students and computer science/mathematics majors is 1.71:1. Thus, computer professionals make a query more efficiently than the business users. Furthermore, during all four query routines, the statistically significant differences are illustrated with respect to the choice of the normalized form scheme, the user's cognitive style, and the user's academic major.

To summarize, the fourth normal form scheme is easier for a user to use compared with the third normal form scheme with respect to both the execution time and the number of relational operations required to complete a query. This is especially the case when the logic to complete a query seems to differ between the two schemes. The comparison between different user groups illustrates that the computer professional makes a query much more efficiently as compared to the business users. This difference between academic majors illustrates that the prior computer and logic training affects the relative ease of use of the database. The results of the cognitive style comparison illustrate the importance of understanding the user's perception in database design, including the user's cognitive style. Cognitive style has a major impact on the amount of time and number of operations required by the business users to complete a query.

Three conclusions can be made regarding the user's operations during the four query routines. The first conclusion is that the fourth normal form database scheme is the easier database system for users to use. The second conclusion is that computer professionals make a query more efficiently than the business users. The last conclusion is that the user's cognitive style should be considered as a variable during database design since the user's cognitive style has an impact upon the relative ease of use of the database.

# CHAPTER EIGHT

## CONCLUSION

This research has shown that the fourth normal form database scheme is more efficient in terms of machine efficiency measurement and end user operation when compared with the third normal form database model. If the original data scheme contains any multivalued dependencies, the normalization process becomes a crucial issue with regard to database design. If multivalued dependencies are present but not detected within the data model, a substantial amount of data redundancy can be expected to occur. If the normalization process is reduced to the third normal form scheme but not as far as the fourth normal form scheme, the mathematical proof has shown that the memory space utilized by the third normal form scheme increases in size in a multiplicative fashion as the complexity of the data relationship increases. If the fourth normal form scheme is utilized, the memory space will increase in an additive fashion as the complexity of the data entities increases.

Furthermore, during the machine efficiency measurements, the fourth normal form scheme performed more efficiently than the third normal form scheme in terms of the memory space utilization and the execution time. These observations were made during the machine simulation that encompassed four query exercises.

Observations regarding the relative ease of use associated with a series of end user queries has shown that the fourth normal form data scheme tended to be an easier database scheme to use. Ease of use was measured by the number of relational operations employed and execution time associated with a completed query routine. Regardless of the normal form used, computer professionals formulated a query more easily than business users.

The user's cognitive style has been shown to be an important factor in measuring the relative ease of use of a query. The user's cognitive style may become a variable of interest during the database design phase. This should allow the database system to perform more efficiently when designed to incorporate the user's cognitive traits.

To summarize, the fourth normal form database scheme is found to be more efficient in terms of machine efficiency measurement and the ease of use when compared with the third normal form database scheme. Consequently, two observations were made. The first observation is that computer professionals made a fast query into a database regardless of the normal form used, while business users' utilization was varied depending upon the normal form used. The second observation is that the user's cognitive style is found to be an important factor in measuring the relative ease of use of a database.

# BIBLIOGRAPHY

# BIBLIOGRAPHY

Aho, A. V., Beeri, C., and Ullman, J. D. "The Theory of Joins in Relational Databases." ACM TODS 4, no. 3 (September, 1979): 297-314.

Alagic, S. Relational Database Technology. New York: Spring-Verlag, 1986

Arora, A. K., and Carlson, C. R. "Normalization Could be Useful." The Computer Journal 27, no. 1 (1984): 57-61.

Atre, S. Data Base: Structured Techniques for Design, Performance, and Management. New York: A Wiley-Interscience Publication, 1980.

Batz, J. "Know Thy Type," Universitas 10, no. 3 (Winter 1985): 23-27.

Beeri, C. "On the Membership problem for Functional and Multivalued Dependencies in Relational Databases." ACM TODS 5, no. 3 (September, 1980): 241-259.

Beeri, C., and Kifer, M. "An Integrated Approach to Logical Design of Relational Database Schemes." ACM TODS 11, no. 2 (June, 1986): 134-158.

Berstein, P. A. "Synthesizing Third Normal Form Relations from Functional Dependencies." ACM TODS 1, no. 4 (December, 1976): 277-298.

Biskup, J. "Inferences of Multivalued Dependencies in Fixed and Undetermined Universes." Theoretical Computer Science 10, no. 1 (January, 1981): 93-106.

Brooks, F. R., and Johnson, R. W. "Self descriptive adjectives assocaited with a jungian personality inventory." Psychological Reports 44, no. 5 (1978): 747-750.

Chen, P. "The entity relationship model: Toward a unified view of data." ACM TODS 11, no. 3 (September, 1986): 9-36.

Codd, E. F. "A relational Model for Large shared Data Banks." Communucations of the ACM 13, no. 1 (1972): 377-387.

Codd, E. F. "Extending the data base relational model to capture more meaning." ACM TODS 4, no. 4 (December, 1979): 397-434.

Date, C. J. An introduction to Database Systems. Readings, Mass.: Addison-Wesley Publishing Company, 1985.

Date, C. J. Relational Database. Readings, Mass.: Addison-Wesley Publishing Company, 1986.

Davis, G. D., and Olson, M. H. Conceptual Foundations, Structure, and Development. New York, New York: McGraw Hill Publishing Co., 1985.

Dickson, G. W., and Wetherbe, J. C. The Management of Information Systems. New York, New York: McGraw Hill Publishing Co., 1985.

Fagin, R. "Multivalued Dependencies and a New Normal Form for Relational Database." ACM TODS 2, no. 3 (Semptember, 1977): 262-278.

Fagin, R. "Normal Forms and Relational Database Operators." ACM SIGMOD (1979): 153-160.

Farin, J., and Nazario, J. "DBMS Basics." InfoSystems 26, no. 6 (June, 1986): 41-46.

Fedorowicz, J. "Database Performance Evaluation in an Indexed File Environment." ACM TODS 12, no. 1 (March 1987): 85-110.

Gallaire, H., and Minker, J. eds. Logic and Databases. New York: Plenum Publishing Company, 1979

Guimaraes, T. "What's Important in a DBMS?" Datamation 27, no. 3 (March 1, 1987): 58.

Haber, A., and Runyon R. P. General Statistics. Philippines: Addison-Wesley Publishing Company, 1969.

Henderson. J. C., and Nuti, P. C. "The influence of Decision Style on Decision Making Behavior." Management Science 26, no. 4 (1980): 371-382.

Hoy, F., and Hellriegel, D. "The Kilmann and Herden Model of Organizational Effectivenss Criteria for Small Business Managers." Academy of Management Journal 25, no. 2 (1980): 308-322.

Huber, G. P. "Cognitive Style as a Basis for Designing MIS and DSS: Much Ado About Nothing?" Management Science 29, no 5. (1983): 567-579.

Inmon, W. H. Effective Database Design. Englewood Cliff, N.J.: Prentice-Hall, 1981.

Jajodia, S. "Recognizing Multivalued Dependencies in Relational Schemas." The Computer Journal 29, no. 5. (1986): 458-459.

Keirsey, D., and Bates, M. Please Understand Me. Del Mar, CA.: Prometheus Nemesis Book, 1978.

Kent W. "A Simple Guide to Five Normal Forms in Relational Database Theory." Communications of the ACM 26, no. 2. (1983): 120-125.

Kroenke, D. Database. Chicago, IL.: Science Research Associates, Inc., 1986.

Lozinskii, E. L. "Construction of Relations in Relational Databases," ACM TODS 5, no. 2 (June, 1980): 208-224.

Maier, D. The Theory of Relational Databases. Rockville, Maryland: Computer Science Press, 1983.

Martin, D. Advanced Database Techniques. Cambridge, Mass.: The MIT Press, 1986.

Martin, J. Managing The Database Environment. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1983.

McFadden, F., and Hoffer J. Data Base Management. Melon Park, CA.: Benjamin Cummings Publishing Company, 1985.

Meadow, C. T. Applied Data Management. New York, New York: John Wiley and Sons, 1976.

Perkinson, R. Data Analysis The Key to Data Base Design. Wellesley, Mass.: QED Inofrmation Sciences, 1984.

Putnam, A. O. Management Information Systems. Boston, Mass.: Herman Publshing, Inc., 1977.

Ramaprassd, A. "Cognitive Process as a Basis for MIS and DSS Design." Management Science 33, no. 2 (1987): 138-148.

Ramsgard, W. C. Making Systems Work. New York, New York: John Wiley and Sons, 1977.

Ray, A. A. SAS User's Guide: Statistics. Cary, North Carolina: SAS Institute, Inc, 1982.

Rissanen, J. "Independent Components of Relations." ACM TODS 2, no. 4 (December 1977): 317-325.

Robey, D. and Taggart, W. "Measuring Manager's Minds." Academy of Management Journal 24, no. 5 (1981): 375-382.

Robey, D. "Cognitive Style as DSS Design: A Comments on Huber's Paper." Management Science 29, no.5 (1983): 580-582.

Schmidt, J. W., and Brodie, M. L. Relational DataBase Systems. New York: Spring Verlag, 1983.

Snedcor, G. W., and Cochran, W. G. Statistical Methods. Ames, Iowa: The Iowa State University Press, 1978.

Taggart, W., and Robey D. "Minds and Managers." Academy of Management Review 6, no. 2 (1981): 187-195.

Teorey, T. J., Fry, J. P. Design of Database Structures. Englewood Cliffs: Prentice-Hall, Inc., 1982

Teorey, T. J., Yang, D., and Fry, J. P. "A Logical Design Methodology for Relational Databases Using the Extended Entity-Relationship Model," Computing Survey 18, no 2 (June 1986): 197-222.

Thierauf, R. J. A Manager's Complete Guide to Effective Information Systems. New York, New York: The Free Press, 1983.

Tscichritzis, D. S., and Lochovsky, F. H. Database Management Systems. New York: Academic Press, 1977.

Tscichritzis, D. S., and Lochvsky, F. H. Data Models. Englewood Cliffs: Prentice-Hall, 1982.

Ullman, J. D. Principles of Database Systems. Rockville, Maryland: Computer Science Press, 1982.

Vesely, E. G. The Practitioner's Blueprint for Logical and Physical Database Design. Englewood Cliffs, N.J.: Prentice-Hall, 1986.

Whetherbe, J. C. Systems Analysis and Design: Traditional, Structured, and Advanced Concepts and Techniques. St. Paul, Minnesota: West Publishing Co., 1984.

Wong, E., and Youssefi, K. "Decomposition - A Strategy for Query Processing." ACM TODS 1, no. 3, (September 1976): 224-241.

Wood, D. "Relational Systems meet the Real World." Data Management 12, no. 7. July, 1985.

Daniolo, C., and Melkanoff, M.A. "On the Design of Relational Database Schemata," ACM TODS 6, no. 1, (March, 1981): 1-47.

# APPENDICES

# APPENDIX A

## THE QUERY ACCESS PATHS FOR THE FOUR QUERY ROUTINES

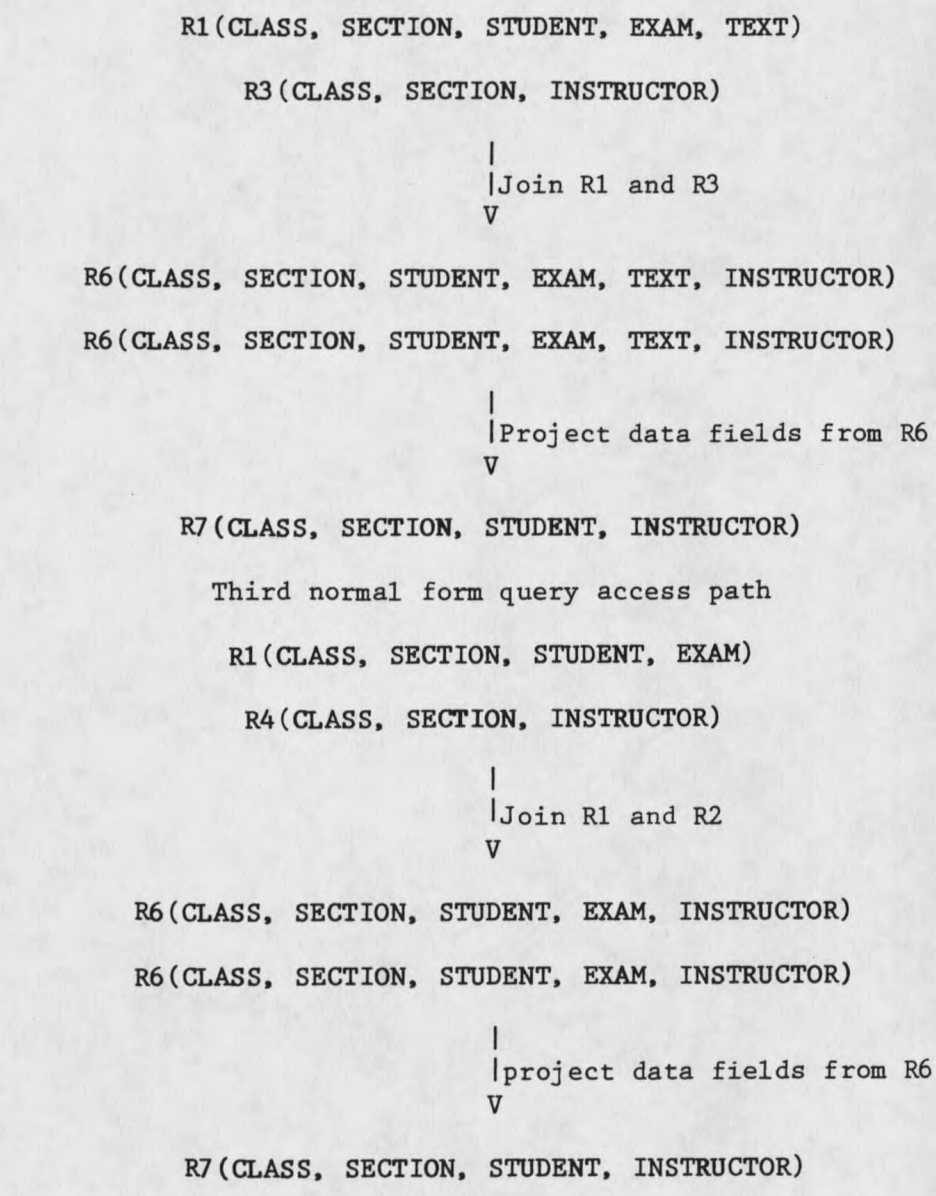



Figure 9. The Query Acess Path for the First Query Routine.

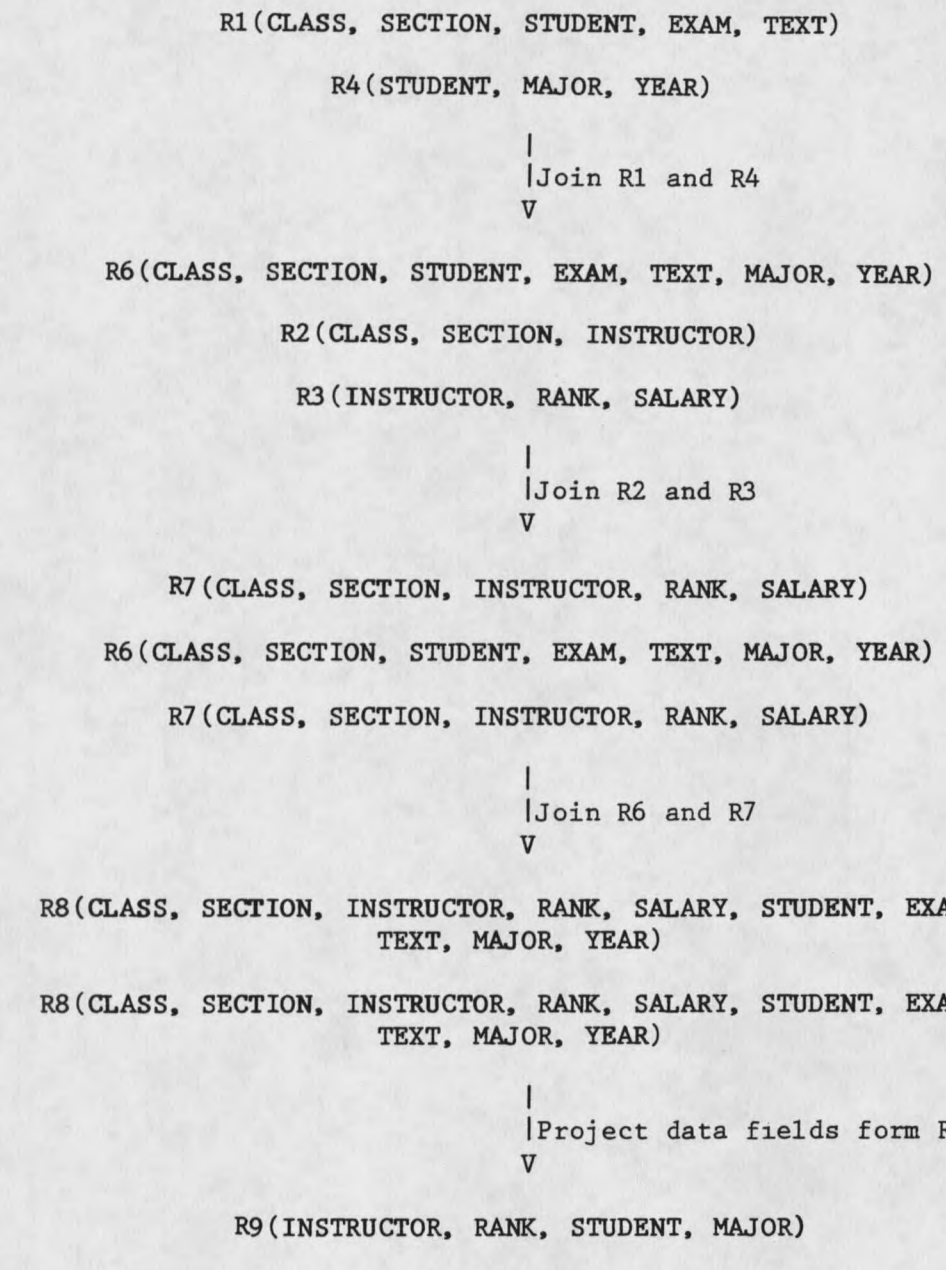
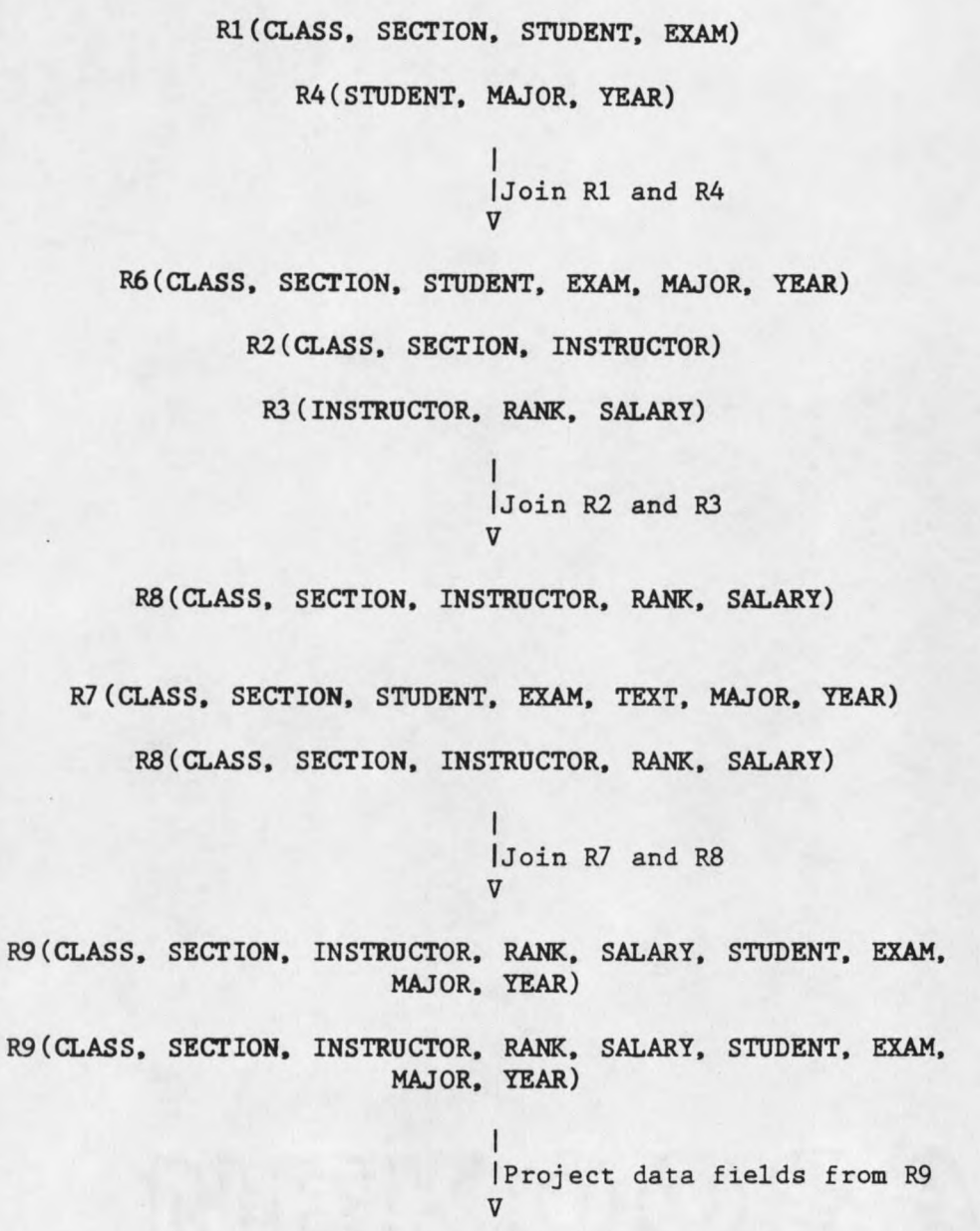



Figure 10. The Query Access Paths for the Second Query Routine.

Figure 10. Continued




Fourth normal form query access path

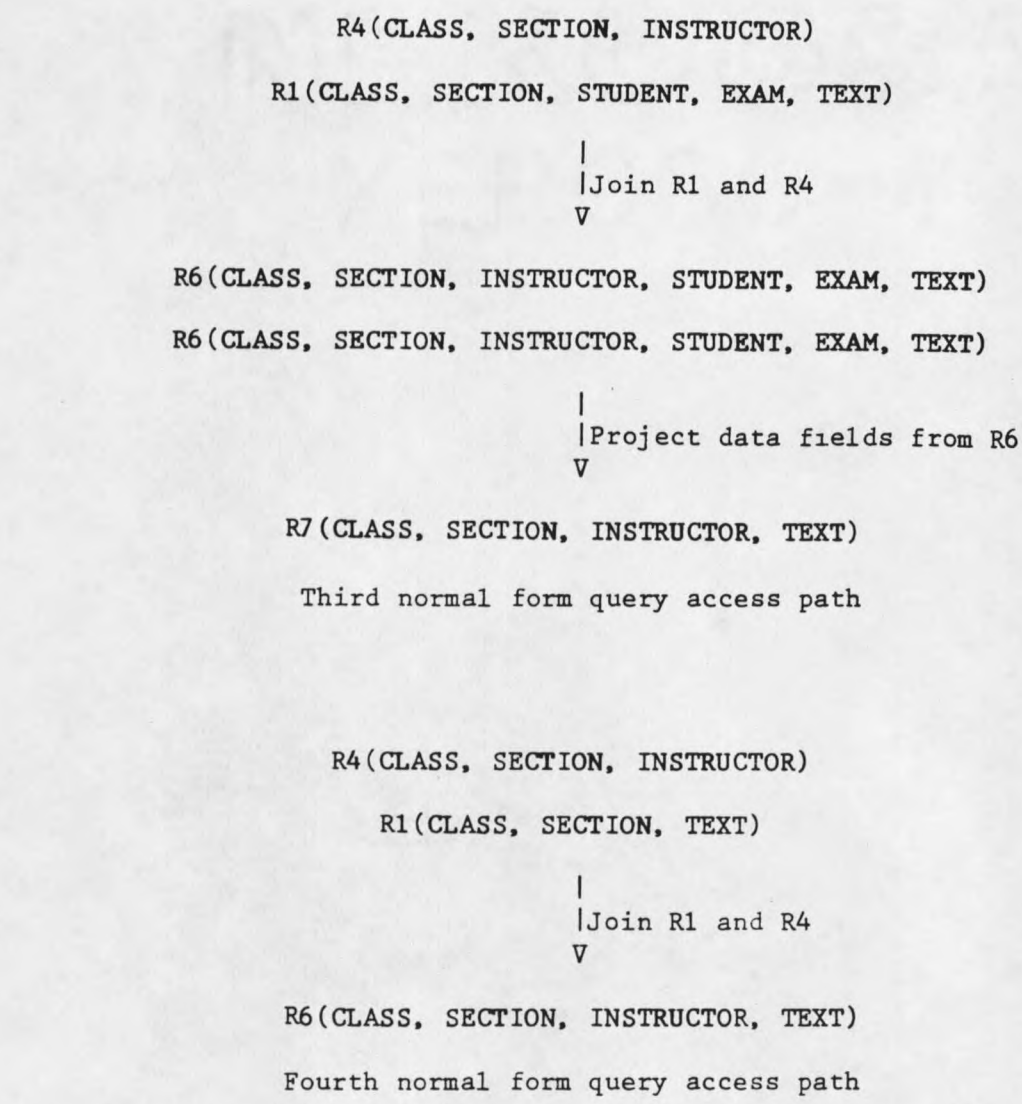



Figure 11. The Query Access Paths for the Third Query Routine.

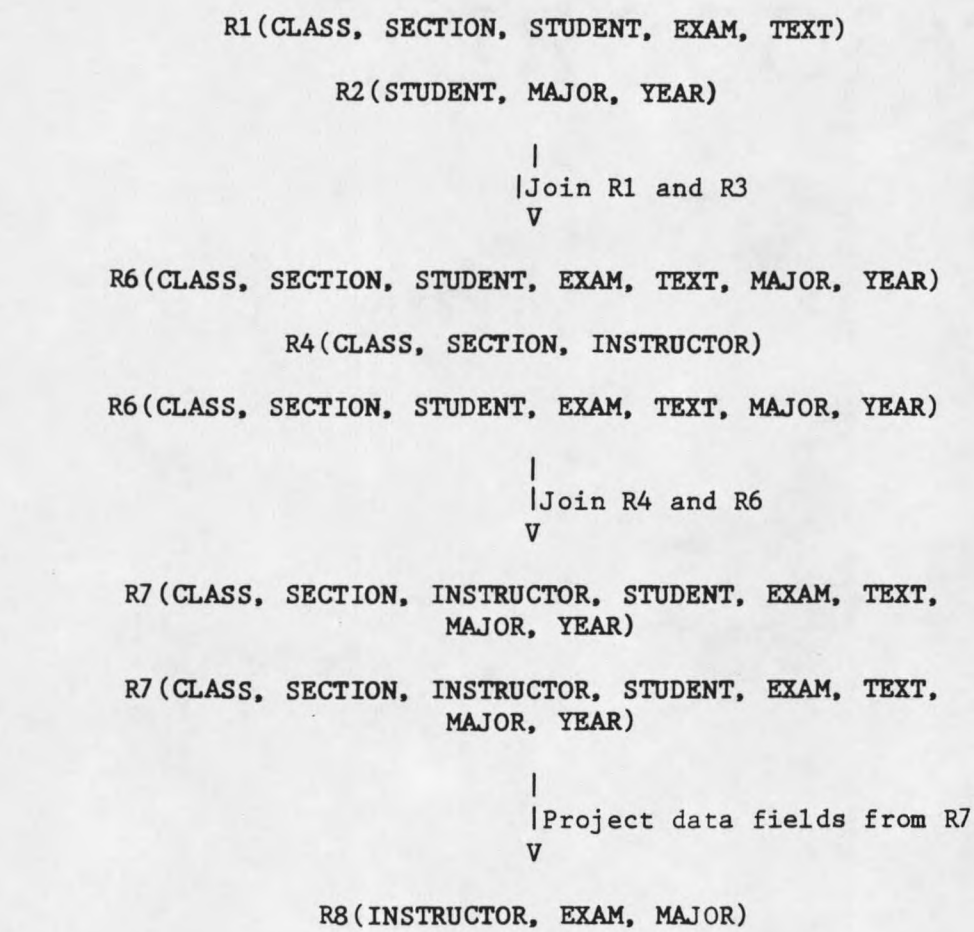
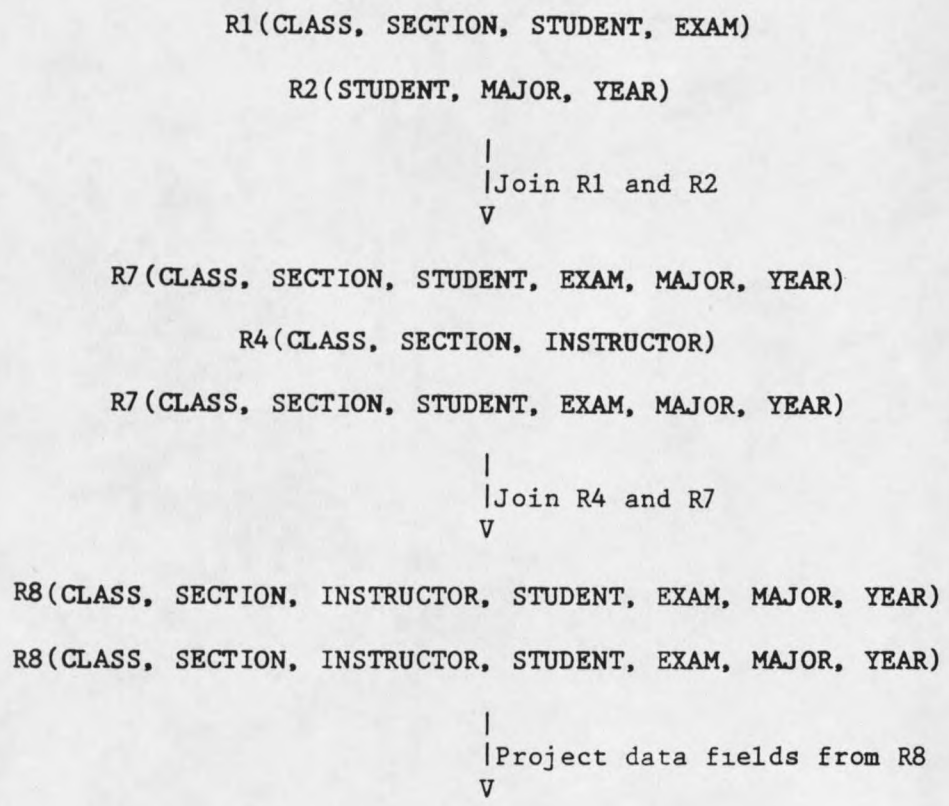



Figure 12. The Query Access Paths for the Fourth Query Routine.

Figure 12. Continued.




Fourth normal form query access path.

# APPENDIX B

## MEMORY ALLOCATIONS FOR THE FOUR QUERY ROUTINES

Table 60. Space Allocated and Used by the Third and the Fourth Normal Form Schemes During the First Query Routine.

| | | Spaces* | | | | Ratios* | | | |
|---|---|---|---|---|---|---|---|---|---|
| M | N | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 1 | 1 | 6413 | 12400 | 5593 | 13000 | 1.14 | 0.95 | 0.51 | 0.43 |
| 1 | 2 | 10163 | 17400 | 5773 | 13600 | 1.76 | 1.27 | 0.58 | 0.42 |
| 1 | 3 | 13914 | 22400 | 5953 | 14200 | 2.33 | 1.57 | 0.62 | 0.41 |
| 1 | 4 | 17663 | 27400 | 6133 | 14800 | 2.87 | 1.85 | 0.64 | 0.41 |
| 1 | 5 | 21413 | 32400 | 6313 | 15400 | 3.39 | 2.10 | 0.66 | 0.40 |
| 2 | 1 | 10163 | 17400 | 8703 | 18000 | 1.16 | 0.96 | 0.58 | 0.48 |
| 2 | 2 | 17663 | 27400 | 8883 | 18600 | 1.98 | 1.47 | 0.64 | 0.47 |
| 2 | 3 | 25163 | 37400 | 9063 | 19200 | 2.77 | 1.94 | 0.67 | 0.47 |
| 2 | 4 | 32663 | 47400 | 11093 | 19800 | 2.94 | 2.39 | 0.68 | 0.56 |
| 2 | 5 | 40163 | 57400 | 11273 | 20400 | 3.56 | 2.81 | 0.69 | 0.55 |
| 3 | 1 | 13913 | 22400 | 11093 | 23000 | 1.25 | 0.97 | 0.62 | 0.48 |
| 3 | 2 | 25163 | 37400 | 11273 | 23600 | 2.23 | 1.58 | 0.67 | 0.47 |
| 3 | 3 | 36413 | 52400 | 11453 | 24200 | 3.17 | 2.16 | 0.69 | 0.47 |
| 3 | 4 | 47663 | 67400 | 11633 | 24800 | 4.09 | 2.71 | 0.70 | 0.46 |
| 3 | 5 | 58913 | 82400 | 11813 | 25400 | 4.98 | 3.24 | 0.71 | 0.46 |
| 4 | 1 | 17663 | 27400 | 13843 | 28000 | 1.27 | 0.97 | 0.64 | 0.49 |
| 4 | 2 | 32663 | 47400 | 14023 | 28600 | 2.32 | 1.65 | 0.68 | 0.49 |
| 4 | 3 | 47663 | 67400 | 14203 | 29200 | 3.35 | 2.30 | 0.70 | 0.48 |
| 4 | 4 | 62663 | 87400 | 14383 | 29800 | 4.35 | 2.93 | 0.71 | 0.48 |
| 4 | 5 | 77663 | 107400 | 14563 | 30400 | 5.33 | 3.53 | 0.72 | 0.47 |
| 5 | 1 | 21413 | 32400 | 16593 | 33000 | 1.29 | 0.98 | 0.66 | 0.50 |
| 5 | 2 | 40163 | 57400 | 16773 | 33600 | 2.39 | 1.70 | 0.69 | 0.49 |
| 5 | 3 | 58913 | 82400 | 16953 | 34200 | 3.47 | 2.40 | 0.71 | 0.49 |
| 5 | 4 | 77663 | 107400 | 17133 | 34800 | 4.53 | 3.08 | 0.72 | 0.49 |
| 5 | 5 | 96413 | 132400 | 17313 | 35400 | 5.56 | 3.74 | 0.72 | 0.48 |

Space 1 space actually used by the third normal form scheme
Space 2 space physically allocated by the third normal form scheme
Space 3 space actually used by the fourth normal form scheme
Space 4 space physically allocated by the fourth normal form scheme
Ratio 1 relative actual space utilized by the third and fourth normal form schemes
Ratio 2 relative physical space allocated by the third and fourth normal form schemes
Ratio 3 relative usage of space by the third normal form scheme
Ratio 4 relative usage of space by the fourth normal form scheme

Table 61. Space Allocated and Used by the Third and the Fourth Normal Form Schemes During the Second Query Routine.

| | | Spaces* | | | | Ratios* | | | |
|---|---|---|---|---|---|---|---|---|---|
| M | N | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 1 | 1 | 9220 | 15500 | 8050 | 16100 | 1.14 | 0.96 | 0.59 | 0.50 |
| 1 | 2 | 15370 | 23000 | 8230 | 16700 | 1.86 | 1.37 | 0.66 | 0.49 |
| 1 | 3 | 21520 | 30500 | 8410 | 17300 | 2.55 | 1.76 | 0.70 | 0.48 |
| 1 | 4 | 27670 | 38000 | 8590 | 17900 | 3.22 | 2.12 | 0.72 | 0.47 |
| 1 | 5 | 39970 | 45500 | 8770 | 18500 | 4.55 | 2.45 | 0.87 | 0.47 |
| 2 | 1 | 15370 | 23000 | 12850 | 23600 | 1.19 | 0.97 | 0.66 | 0.54 |
| 2 | 2 | 27670 | 38000 | 13030 | 24200 | 2.12 | 1.57 | 0.72 | 0.53 |
| 2 | 3 | 39970 | 53000 | 13210 | 24800 | 3.02 | 2.13 | 0.75 | 0.53 |
| 2 | 4 | 52270 | 68000 | 13390 | 25400 | 3.90 | 2.67 | 0.76 | 0.52 |
| 2 | 5 | 64570 | 83000 | 13570 | 26000 | 4.75 | 3.19 | 0.77 | 0.52 |
| 3 | 1 | 21520 | 30500 | 17650 | 31100 | 1.21 | 0.98 | 0.70 | 0.56 |
| 3 | 2 | 39970 | 53000 | 17830 | 31700 | 2.24 | 1.67 | 0.75 | 0.56 |
| 3 | 3 | 58420 | 75500 | 18010 | 32300 | 3.24 | 2.33 | 0.77 | 0.55 |
| 3 | 4 | 76870 | 98000 | 18190 | 32900 | 4.22 | 2.97 | 0.78 | 0.55 |
| 3 | 5 | 95320 | 120500 | 18370 | 33500 | 5.18 | 3.59 | 0.79 | 0.54 |
| 4 | 1 | 27670 | 38000 | 22450 | 38600 | 1.23 | 0.98 | 0.72 | 0.58 |
| 4 | 2 | 52270 | 68000 | 22630 | 39200 | 2.30 | 1.73 | 0.76 | 0.57 |
| 4 | 3 | 76870 | 98000 | 22810 | 39800 | 3.37 | 2.43 | 0.78 | 0.57 |
| 4 | 4 | 101470 | 128000 | 22990 | 40400 | 4.41 | 3.16 | 0.79 | 0.56 |
| 4 | 5 | 126070 | 158000 | 23170 | 41000 | 5.44 | 3.85 | 0.79 | 0.56 |
| 5 | 1 | 33820 | 45500 | 27250 | 46100 | 1.24 | 0.98 | 0.74 | 0.59 |
| 5 | 2 | 64570 | 83000 | 27430 | 46700 | 2.35 | 1.77 | 0.77 | 0.58 |
| 5 | 3 | 95320 | 120500 | 27610 | 47300 | 3.45 | 2.54 | 0.79 | 0.58 |
| 5 | 4 | 126070 | 158000 | 27790 | 47900 | 4.53 | 3.29 | 0.79 | 0.58 |
| 5 | 5 | 156820 | 195500 | 27970 | 48500 | 5.60 | 4.03 | 0.80 | 0.57 |

Space 1 space actually used by the third normal form scheme
Space 2 space physically allocated by the third normal form scheme
Space 3 space actually used by the fourth normal form scheme
Space 4 space physically allocated by the fourth normal form scheme
Ratio 1 relative actual space utilized by the third and fourth normal form schemes
Ratio 2 relative physical space allocated by the third and fourth normal form schemes
Ratio 3 relative usage of space by the third normal form scheme
Ratio 4 relative usage of space by the fourth normal form scheme

Table 62. Space Allocated and Used by the Third and Fourth Normal Form Schemes During the Third Query Routine.

| | | Spaces* | | | | Ratios* | | | |
|---|---|---|---|---|---|---|---|---|---|
| M | N | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 1 | 1 | 5463 | 10500 | 3018 | 8600 | 1.81 | 1.22 | 0.52 | 0.35 |
| 1 | 2 | 9513 | 16100 | 3498 | 9800 | 2.71 | 1.64 | 0.59 | 0.35 |
| 1 | 3 | 13563 | 21700 | 3978 | 11000 | 3.40 | 1.97 | 0.62 | 0.36 |
| 1 | 4 | 17613 | 27300 | 4458 | 12200 | 3.95 | 2.23 | 0.64 | 0.36 |
| 1 | 5 | 21663 | 32900 | 4938 | 13400 | 4.38 | 2.43 | 0.65 | 0.36 |
| 2 | 1 | 9218 | 15500 | 4143 | 11100 | 2.22 | 1.39 | 0.59 | 0.37 |
| 2 | 2 | 17013 | 26100 | 4623 | 12300 | 3.68 | 2.12 | 0.65 | 0.37 |
| 2 | 3 | 24813 | 36700 | 5103 | 13500 | 4.86 | 2.71 | 0.67 | 0.37 |
| 2 | 4 | 32613 | 47300 | 5583 | 14700 | 5.85 | 3.21 | 0.68 | 0.37 |
| 2 | 5 | 40413 | 57900 | 6063 | 15900 | 6.64 | 3.64 | 0.69 | 0.38 |
| 3 | 1 | 12963 | 20500 | 5268 | 13600 | 2.46 | 1.50 | 0.63 | 0.38 |
| 3 | 2 | 24513 | 36100 | 5748 | 14800 | 4.26 | 2.43 | 0.67 | 0.38 |
| 3 | 3 | 36063 | 51700 | 6228 | 16000 | 5.79 | 3.23 | 0.69 | 0.38 |
| 3 | 4 | 47613 | 67300 | 6708 | 17200 | 7.09 | 3.91 | 0.70 | 0.39 |
| 3 | 5 | 59163 | 82900 | 7188 | 18400 | 8.23 | 4.50 | 0.71 | 0.39 |
| 4 | 1 | 16713 | 25500 | 6393 | 16100 | 2.61 | 1.58 | 0.65 | 0.39 |
| 4 | 2 | 32013 | 46100 | 6873 | 17300 | 4.65 | 2.66 | 0.69 | 0.39 |
| 4 | 3 | 47313 | 66700 | 7353 | 18500 | 6.43 | 3.60 | 0.70 | 0.39 |
| 4 | 4 | 62613 | 87300 | 7833 | 19700 | 7.99 | 4.43 | 0.71 | 0.39 |
| 4 | 5 | 77913 | 107900 | 8313 | 20900 | 9.37 | 5.16 | 0.72 | 0.39 |
| 5 | 1 | 20493 | 30500 | 7518 | 18600 | 2.72 | 1.63 | 0.67 | 0.40 |
| 5 | 2 | 39513 | 56100 | 7998 | 19800 | 4.94 | 2.83 | 0.70 | 0.40 |
| 5 | 3 | 58563 | 81700 | 8478 | 21000 | 6.90 | 3.89 | 0.71 | 0.40 |
| 5 | 4 | 77613 | 107300 | 8958 | 22200 | 8.66 | 4.83 | 0.72 | 0.40 |
| 5 | 5 | 96663 | 132900 | 9438 | 23400 | 10.24 | 5.67 | 0.72 | 0.40 |

Space 1 space actually used by the third normal form scheme
Space 2 space physically allocated by the third normal form scheme
Space 3 space actually used by the fourth normal form scheme
Space 4 space physically allocated by the fourth normal form scheme

Ratio 1 relative actual space utilized by the third and fourth normal form schemes
Ratio 2 relative physical space allocated by the third and fourth normal form schemes
Ratio 3 relative usage of space by the third normal form scheme
Ratio 4 relative usage of space by the fourth normal form scheme

Table 63. Space Allocated and Used by the Third and Fourth Normal Form Schemes During the Fourth Query Routine.

| | | Spaces* | | | | Ratios* | | | |
|---|---|---|---|---|---|---|---|---|---|
| M | N | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 1 | 1 | 7378 | 13000 | 6058 | 13600 | 1.21 | 0.95 | 0.56 | 0.44 |
| 1 | 2 | 13103 | 20500 | 6238 | 14200 | 2.10 | 1.44 | 0.63 | 0.43 |
| 1 | 3 | 18828 | 28000 | 6418 | 14800 | 2.93 | 1.89 | 0.67 | 0.43 |
| 1 | 4 | 24553 | 35500 | 6598 | 15400 | 3.72 | 2.30 | 0.69 | 0.42 |
| 1 | 5 | 30278 | 43000 | 6778 | 16000 | 4.46 | 2.68 | 0.70 | 0.42 |
| 2 | 1 | 13343 | 21100 | 10523 | 21700 | 1.26 | 0.97 | 0.63 | 0.48 |
| 2 | 2 | 24793 | 36100 | 10703 | 22300 | 2.31 | 1.61 | 0.68 | 0.47 |
| 2 | 3 | 36243 | 51100 | 10883 | 22900 | 3.33 | 2.23 | 0.70 | 0.47 |
| 2 | 4 | 47693 | 66100 | 11063 | 23500 | 4.31 | 2.81 | 0.72 | 0.47 |
| 2 | 5 | 58143 | 81100 | 11243 | 24100 | 5.17 | 3.36 | 0.71 | 0.46 |
| 3 | 1 | 19308 | 29200 | 14988 | 29800 | 1.28 | 0.97 | 0.66 | 0.50 |
| 3 | 2 | 36483 | 51700 | 15168 | 30400 | 2.40 | 1.70 | 0.70 | 0.49 |
| 3 | 3 | 53658 | 74200 | 15348 | 31000 | 3.49 | 2.39 | 0.72 | 0.49 |
| 3 | 4 | 70833 | 96700 | 15528 | 31600 | 4.56 | 3.06 | 0.73 | 0.49 |
| 3 | 5 | 88008 | 127300 | 15708 | 32200 | 5.60 | 3.95 | 0.69 | 0.48 |
| 4 | 1 | 25273 | 37300 | 19453 | 37900 | 1.29 | 0.98 | 0.67 | 0.51 |
| 4 | 2 | 48173 | 67300 | 19633 | 38500 | 2.45 | 1.74 | 0.71 | 0.50 |
| 4 | 3 | 71073 | 97300 | 19813 | 39100 | 3.58 | 2.48 | 0.73 | 0.50 |
| 4 | 4 | 93973 | 127300 | 19993 | 39700 | 4.70 | 3.20 | 0.73 | 0.50 |
| 4 | 5 | 116873 | 157300 | 20173 | 40300 | 5.79 | 3.90 | 0.74 | 0.50 |
| 5 | 1 | 31238 | 45400 | 23918 | 46000 | 1.30 | 0.98 | 0.68 | 0.51 |
| 5 | 2 | 59863 | 82900 | 24098 | 46600 | 2.48 | 1.77 | 0.72 | 0.51 |
| 5 | 3 | 88488 | 120400 | 24458 | 47200 | 3.61 | 2.55 | 0.73 | 0.51 |
| 5 | 4 | 117113 | 157900 | 24458 | 47800 | 4.78 | 3.30 | 0.74 | 0.51 |
| 5 | 5 | 145738 | 195400 | 24638 | 48400 | 5.91 | 4.03 | 0.74 | 0.50 |

Space 1 space actually used by the third normal form scheme
Space 2 space physically allocated by the third normal form scheme
Space 3 space actually used by the fourth normal form scheme
Space 4 space physically allocated by the fourth normal form scheme
Ratio 1 relative actual space utilized by the third and fourth normal form schemes
Ratio 2 relative physical space allocated by the third and fourth normal form schemes
Ratio 3 relative usage of space by the third normal form scheme
Ratio 4 relative usage of space by the fourth normal form scheme

# APPENDIX C

## THE EXECUTION TIMES FOR THE FOUR QUERY ROUTINES

Table 64. The Execution Time Used by Both the Third and the Fourth Normal Form Schemes During the First Query Routine.

| M | N | third | fourth | ratio |
|---|---|---|---|---|
| 1 | 1 | 1494 | 1357 | 1.10091 |
| 1 | 2 | 2291 | 1359 | 1.68580 |
| 1 | 3 | 2658 | 1359 | 1.95585 |
| 1 | 4 | 3488 | 1362 | 2.56094 |
| 1 | 5 | 5169 | 1354 | 3.81758 |
| 2 | 1 | 2054 | 1756 | 1.16970 |
| 2 | 2 | 3449 | 1757 | 1.96301 |
| 2 | 3 | 4751 | 1757 | 2.70404 |
| 2 | 4 | 6184 | 1758 | 3.51763 |
| 2 | 5 | 7464 | 1759 | 4.24332 |
| 3 | 1 | 2691 | 2202 | 1.22207 |
| 3 | 2 | 4872 | 2202 | 2.21253 |
| 3 | 3 | 6866 | 2204 | 3.11525 |
| 3 | 4 | 9057 | 2205 | 4.10748 |
| 3 | 5 | 11315 | 2206 | 5.12919 |
| 4 | 1 | 3487 | 2709 | 1.28719 |
| 4 | 2 | 6701 | 2710 | 2.47269 |
| 4 | 3 | 9568 | 2711 | 3.52932 |
| 4 | 4 | 12578 | 2712 | 4.63791 |
| 4 | 5 | 15472 | 2714 | 5.70081 |
| 5 | 1 | 4449 | 3201 | 1.38988 |
| 5 | 2 | 8365 | 3201 | 2.61325 |
| 5 | 3 | 11919 | 3203 | 3.72120 |
| 5 | 4 | 15637 | 3204 | 4.88046 |
| 5 | 5 | 19421 | 3278 | 5.92465 |

M the number of examinations (EXAM)
N the number of textbooks (TEXT)
third the execution time required for the third normal form to complete the first query routine
fourth the execution time required for the fourth normal form to complete the first query routine
ratio a division of third by fourth

Table 65. The Execution Time Used by Both the Third and the Fourth Normal Form Schemes During the Second Query Routine.

| M | N | third | fourth | ratio |
|---|---|---|---|---|
| 1 | 1 | 5081 | 4949 | 1.02667 |
| 1 | 2 | 5283 | 4951 | 1.06706 |
| 1 | 3 | 5501 | 4951 | 1.11109 |
| 1 | 4 | 5702 | 4952 | 1.15145 |
| 1 | 5 | 5892 | 4955 | 1.18910 |
| 2 | 1 | 9354 | 7962 | 1.17483 |
| 2 | 2 | 9634 | 7963 | 1.20985 |
| 2 | 3 | 9931 | 7964 | 1.24699 |
| 2 | 4 | 11232 | 7965 | 1.41017 |
| 2 | 5 | 11433 | 7968 | 1.43486 |
| 3 | 1 | 15320 | 10088 | 1.51864 |
| 3 | 2 | 15531 | 10090 | 1.53925 |
| 3 | 3 | 15782 | 10092 | 1.56381 |
| 3 | 4 | 15992 | 10094 | 1.58431 |
| 3 | 5 | 16210 | 10094 | 1.60590 |
| 4 | 1 | 19432 | 12088 | 1.60754 |
| 4 | 2 | 19678 | 12090 | 1.62763 |
| 4 | 3 | 19943 | 12091 | 1.64941 |
| 4 | 4 | 20134 | 12091 | 1.66521 |
| 4 | 5 | 20346 | 12094 | 1.68232 |
| 5 | 1 | 24683 | 14764 | 1.67184 |
| 5 | 2 | 24892 | 14765 | 1.68588 |
| 5 | 3 | 25024 | 14766 | 1.69470 |
| 5 | 4 | 25302 | 14768 | 1.71330 |
| 5 | 5 | 25602 | 14770 | 1.73338 |

M the number of examinations (EXAM)
N the number of textbooks (TEXT)
third the execution time required for the third normal form to complete the second query routine
fourth the execution time required for the fourth normal form to complete the second query routine
ratio a division of third by fourth

Table 66. The Execution Time Used by Both the Third and the Fourth Normal Form Schemes During the Third Query Routine.

| M | N | third | fourth | ratio |
|---|---|---|---|---|
| 1 | 1 | 1340 | 678 | 1.9764 |
| 1 | 2 | 1829 | 681 | 2.6857 |
| 1 | 3 | 2037 | 685 | 2.9732 |
| 1 | 4 | 2609 | 692 | 3.7703 |
| 1 | 5 | 2955 | 698 | 4.2352 |
| 2 | 1 | 1653 | 682 | 2.4283 |
| 2 | 2 | 2532 | 683 | 3.7072 |
| 2 | 3 | 3351 | 685 | 4.8920 |
| 2 | 4 | 4306 | 687 | 6.2678 |
| 2 | 5 | 4993 | 692 | 7.2153 |
| 3 | 1 | 2054 | 683 | 3.0073 |
| 3 | 2 | 3499 | 685 | 5.1080 |
| 3 | 3 | 4647 | 687 | 6.7642 |
| 3 | 4 | 5981 | 689 | 8.6807 |
| 3 | 5 | 7542 | 694 | 10.8674 |
| 4 | 1 | 2735 | 681 | 4.0162 |
| 4 | 2 | 4383 | 683 | 6.4173 |
| 4 | 3 | 6448 | 684 | 9.4269 |
| 4 | 4 | 8184 | 686 | 11.9300 |
| 4 | 5 | 10107 | 687 | 14.7118 |
| 5 | 1 | 3257 | 683 | 4.7687 |
| 5 | 2 | 5547 | 686 | 8.0860 |
| 5 | 3 | 7925 | 688 | 11.5189 |
| 5 | 4 | 10282 | 689 | 14.9231 |
| 5 | 5 | 12602 | 692 | 18.2110 |

M the number of examinations (EXAM)
N the number of textbooks (TEXT)
third the execution time required for the third normal form to complete query one
fourth the execution time required for the fourth normal form to complete query one

Table 67. The Execution Time Used by Both the Third and Fourth Normal Form Schemes During the Fourth Query Routine.

| M | N | third | fourth | ratio |
|---|---|---|---|---|
| 1 | 1 | 5081 | 4879 | 1.0414 |
| 1 | 2 | 5169 | 4879 | 1.0594 |
| 1 | 3 | 5309 | 4881 | 1.0877 |
| 1 | 4 | 5685 | 4882 | 1.1645 |
| 1 | 5 | 5949 | 4885 | 1.2178 |
| 2 | 1 | 9053 | 7783 | 1.1632 |
| 2 | 2 | 9123 | 7784 | 1.1720 |
| 2 | 3 | 9221 | 7785 | 1.1845 |
| 2 | 4 | 9321 | 7786 | 1.1971 |
| 2 | 5 | 9501 | 7788 | 1.2200 |
| 3 | 1 | 13892 | 11619 | 1.1956 |
| 3 | 2 | 14210 | 11621 | 1.2228 |
| 3 | 3 | 14890 | 11622 | 1.2812 |
| 3 | 4 | 15492 | 11623 | 1.3329 |
| 3 | 5 | 15910 | 11624 | 1.3687 |
| 4 | 1 | 18212 | 15632 | 1.1650 |
| 4 | 2 | 18934 | 15634 | 1.2111 |
| 4 | 3 | 19230 | 15634 | 1.2300 |
| 4 | 4 | 19578 | 15635 | 1.2522 |
| 4 | 5 | 19902 | 15637 | 1.2728 |
| 5 | 1 | 23421 | 19186 | 1.2207 |
| 5 | 2 | 23893 | 19187 | 1.2453 |
| 5 | 3 | 24342 | 19190 | 1.2685 |
| 5 | 4 | 24790 | 19193 | 1.2916 |
| 5 | 5 | 25101 | 19195 | 1.3077 |

M the number of examinations (EXAM)
N the number of textbooks (TEXT)
third the execution time required for the third normal form to complete query one
fourth the execution time required for the fourth normal form to complete query one
ratio a division of third by fourth

# APPENDIX D

# QUESTIONNAIRE

# QUESTIONNAIRE

PART I: We would like to know a little bit about your background.

1. Have you worked on a large (mainframe) computer?

   A. often B. sometimes C. never

2. Have you worked with a microcomputer before?

   A. often B. sometimes C. never

3. Do you own a personal computer? A. Yes B. No

   If yes, what type? (please specify) _____

4. Do your parents have a personal computer at home? A. Yes B. No

5. For how many years have you used a large (mainframe) computer?

A. none
B. less than 6 months
C. more than 6 months but less than 1 year
D. more than 1 year but less than 2 year
E. more than 2 years

6. For how many years have you used a microcomputer?

A. none
B. less than 6 months
C. more than 6 months but less than 1 year
D. more than 1 year but less than 2 year
E. more than 2 years

7. How would you rate your skills with the personal computer?

   A. Excellent B. Good C. Fair D. Weak E. None at all

8. Have you ever used a database program? A. Yes B. No

   a. If yes, what program(s) did you use? _____

      _____

      _____

Check any of the following operations that you may have performed:

| | | |
|---|---|---|
| 9. search for names | A. Yes | B. No |
| 10. create a file | A. Yes | B. No |
| 11. combine two files | A. Yes | B. No |
| 12. create a report | A. Yes | B. No |
| 13. combine two files in a report | A. Yes | B. No |
| 14. print a report | A. Yes | B. No |

15. Have you ever taken a database course? A. Yes B. No

16. How many computer credits have you taken in college?

A. 0 - 4 credits
B. 5 - 8 credits
C. 9 - 12 credits
D. 13 - 16 credits
E. more than 17 credits

17. How many mathematics or logic courses have you taken in college?

A. 0 - 4 credits
B. 5 - 8 credits
C. 9 - 12 credits
D. 13 - 16 credits
E. more than 16 credits

PART II: Now, please, circle your answers to the following questions.

18 Are you more careful about?
A. people's feelings
B. their rights

19 When you have to meet stranger do you find it:

A. something that takes a good deal of effort.
B. pleasant, or at least easy little

20 Does following a schedule:
A. appeal to you

B. cramp you

21 Do you usually get on better with:
A. imaginative people

29 Is it harder to praise to call someone:
A. a man of vision
B. a man of common sense

30 Do you more often let:
A. your heart rule your head
B. your head rule your heart

31 When you think of some thing you should do or buy, do you:
A. often forget it until much late
B. usually get it down on paper before it escapes you

32 Can you:
A. talk easily to almost anyone for as long as you

B. realistic people

22 Are you naturally:
A. rather quiet and reserved in company
B. a good "mixer"

23 Is it harder for you to adapt to:
A. routine
B. constant change

24 Which of these two is the higher complement:
A. he is a person of real feeling
B. he is consistently reasonable

25 Would you judge yourself to be:
A. more enthusiastic than the average
B. less excitable than the average person

26 In doing something with many other people, does it appeal more to you:
A. to do it in the accepted way
B. to invent a way of your own

27 Are you at your best:
A. when floowing a carefully worked out plan
B. when dealing with the unexpected

28 Do you get more annoyed at:
A. fancy theories
B. people who don't like theories

have to
B. find a lot to say only to certain people or certain conditions

33 Do you think it is a worse fault:
A. to show too much warmth
B. to to be unsympathetic

34 If you were a teacher, would you rather teach:
A. courses involving theory
B. fact courses

35 When it is settled well in advance that you will do a certain thing at a certain time, do you find it:
A. nice to be able to plan accordingly
B. a little plesant to be tied down

36 Can the new people you meet tell what you are interested in:
A. right away
B. only after they really get know you

37 In your daily work, do you:
A. rather enjoy a emergency that makes you work against time
B. usually plan your work so you won't need to

38 In a large group, do you more often:
A. introduce others
B. get introduced

Part III. Which word in the following pair appeal to your more?

| | | |
|---|---|---|
| 39 | A. compassion | B. foresight |
| 40 | A. punctual | B. leisurely |
| 41 | A. justice | B. mercy |
| 42 | A. production | B. design |
| 43 | A. gentle | B. firm |
| 44 | A. uncritical | B. critical |
| 45 | A. calm | B. lively |

46 A. theory B. experience
47 A. literal B. figurative
48 A. imaginative B. matter-of-fact

49. Major

A. Business management
B. Business marketing/ finance
C. Business information system/ education
D. Business other field
E. Computer science

50. Year

A. Freshman
B. Sophomore
C. Junior
D. Senior
E. Graduate

51. Sex

A. Male
B. Female

52. Age

A. less than 19 years old
B. 20 - 22 years old
C. 23 - 24 years old
D. 25 - 26 years old
E. older than 27 years old

53. Grade point average

A. less than 2.00
B. 2.01 - 2.50
C. 2.51 - 3.00
D. 3.01 - 3.50
E. 3.51 - 4.00

# APPENDIX E

## DATA

(Data Labels and Raw Data)

| POSITION | DESCRIPTION | CODE |
|---|---|---|
| 1-3 | ID NUMBER | 001-999 |
| 11 | QUESTION #1 IN QUESTIONNAIRE (see Appendix D) | 1. often<br>2. sometimes<br>3. never |
| 12 | QUESTION #2 IN QUESTIONNAIRE | 1. often<br>2. sometimes<br>3. never |
| 13 | QUESTION #3 IN QUESTIONNAIRE | 1. Yes<br>2. No |
| 14 | QUESTION #4 IN QUESTIONNAIRE | 1. Yes<br>2. No |
| 15 | QUESTION #5 IN QUESTIONNAIRE | 1. none<br>2. less than 6 months<br>3. more than 6 months but less than 1 year<br>4. more than 1 year but less than 2 years<br>5. more than 2 years |
| 16 | QUESTION #6 IN QUESTIONNAIRE | 1. none<br>2. less than 6 months<br>3. more than 6 months but less than 1 year<br>4. more than 1 year but less than 2 years<br>5. more than 2 years |
| 17 | QUESTION #7 IN QUESTIONNAIRE | 1. Excellent<br>2. Good<br>3. Fair<br>4. Weak<br>5. None at all |
| 18 | QUESTION #8 IN QUESTIONNAIRE | 1. Yes<br>2. No |
| 19 | QUESTION #9 IN QUESTIONNAIRE | 1. Yes<br>2. No |
| 20 | QUESTION #10 IN QUESTIONNAIRE | 1. Yes<br>2. No |
| 21 | QUESTION #11 IN QUESTIONNAIRE | 1. Yes<br>2. No |
| 22 | QUESTION #12 IN QUESTIONNAIRE | 1. Yes<br>2. No |
| 23 | QUESTION #13 IN QUESTIONNAIRE | 1. Yes<br>2. No |

| | | |
|---|---|---|
| 24 | QUESTION #14 IN QUESTIONNAIRE | 1. Yes<br>2. No |
| 25 | QUESTION #15 IN QUESTIONNAIRE | 1. Yes<br>2. No |
| 26 | QUESTION #16 IN QUESTIONNAIRE | 1. 0-4 credits<br>2. 5-8 credits<br>3. 9-12 credits<br>4. 13-16 credits<br>5. more than 17 credits |
| 27 | QUESTION #17 IN QUESTIONNAIRE | 1. 0-4 credits<br>2. 5-8 credits<br>3. 9-12 credits<br>4. 13-16 credits<br>5. more than 17 credits |
| 28 | QUESTION #18 IN QUESTIONNAIRE | 1. people's feeling<br>2. their rights |
| 29 | QUESTION #19 IN QUESTIONNAIRE | 1. something that takes a good deal of effort<br>2. pleasant, or at least easy little |
| 30 | QUESTION #20 IN QUESTIONNAIRE | 1. appeal to you<br>2. cramp you |
| 31 | QUESTION #21 IN QUESTIONNAIRE | 1. imaginative people<br>2. realistic people |
| 32 | QUESTION #22 IN QUESTIONNAIRE | 1. rather quiet and reserved in company<br>2. a good "mixer" |
| 33 | QUESTION #23 IN QUESTIONNAIRE | 1. routine<br>2. constant change |
| 34 | QUESTION #24 IN QUESTIONNAIRE | 1. he is a person of real feeling<br>2. he is consistently reasonable |
| 35 | QUESTION #25 IN QUESTIONNAIRE | 1. more enthusiastic than the average<br>2. less excitable than the average person |
| 36 | QUESTION #26 IN QUESTIONNAIRE | 1. to do it in the accepted way<br>2. to invent a way of your own |
| 37 | QUESTION #27 IN QUESTIONNAIRE | 1. when floowing a carefully worked out plan<br>2. when dealing with the unexpected |
| 38 | QUESTION #28 IN QUESTIONNAIRE | 1. fancy theories<br>2. people who don't like theories |

| | | |
|---|---|---|
| 39 | QUESTION #29 IN QUESTIONNAIRE | 1. a man of vision<br>2. a man of common sense |
| 40 | QUESTION #30 IN QUESTIONNAIRE | 1. your heart rule your head<br>2. your head rule your heart |
| 41 | QUESTION #31 IN QUESTIONNAIRE | 1. often forget it until much late<br>2. usually get it down on paper before it escapes you |
| 42 | QUESTION #32 IN QUESTIONNAIRE | 1. talk easily to almost anyone for as long as you have to<br>2. find a lot to say only to certain people or certain conditions |
| 43 | QUESTION #33 IN QUESTIONNAIRE | 1. to show too much warmth<br>2. to be unsympathetic |
| 44 | QUESTION #34 IN QUESTIONNAIRE | 1. courses involving theory<br>2. fact courses |
| 45 | QUESTION #35 IN QUESTIONNAIRE | 1. nice to be able to plan accordingly<br>2. a little pleasant to be tied down |
| 46 | QUESTION #36 IN QUESTIONNAIRE | 1. right away<br>2. only after they really get know you |
| 47 | QUESTION #37 IN QUESTIONNAIRE | 1. rather enjoy a emergency that makes you work against time<br>2. usually plan your work so you won't need to |
| 48 | QUESTION #38 IN QUESTIONNAIRE | 1. introduce others<br>2. get introduced |
| 49 | QUESTION #39 IN QUESTIONNAIRE | 1. compassion<br>2. foresight |
| 50 | QUESTION #40 IN QUESTIONNAIRE | 1. punctual<br>2. leisurely |
| 51 | QUESTION #41 IN QUESTIONNAIRE | 1. justice<br>2. mercy |
| 52 | QUESTION #42 IN QUESTIONNAIRE | 1. production<br>2. design |
| 53 | QUESTION #43 IN QUESTIONNAIRE | 1. gentle<br>2. firm |
| 54 | QUESTION #44 IN QUESTIONNAIRE | 1. uncritical<br>2. critical |

| | | |
|---|---|---|
| 55 | QUESTION #45 IN QUESTIONNAIRE | 1. calm<br>2. lively |
| 56 | QUESTION #46 IN QUESTIONNAIRE | 1. theory<br>2. experience |
| 57 | QUESTION #47 IN QUESTIONNAIRE | 1. literal<br>2. figurative |
| 58 | QUESTION #48 IN QUESTIONNAIRE | 1. imaginative<br>2. matter-of-fact |
| 59 | QUESTION #49 IN QUESTIONNAIRE | 1. Business management<br>2. Business marketing/ finance<br>3. Business information system/ education<br>4. Business other field<br>5. Computer Science |
| 60 | QUESTION #50 IN QUESTIONNAIRE | 1. Freshman<br>2. Sophomore<br>3. Junior<br>4. Senior<br>5. Graduate |
| 61 | QUESTION #51 IN QUESTIONNAIRE | 1. Male<br>2. Female |
| 62 | QUESTION #52 IN QUESTIONNAIRE | 1. less than 19 years old<br>2. 20-22 years old<br>3. 23-24 years old<br>4. 25-26 years old<br>5. older than 27 years old |
| 63 | QUESTION #53 IN QUESTIONNAIRE | 1. less than 2.00<br>2. 2.01-2.50<br>3. 2.51-3.00<br>4. 3.01-3.50<br>5. 3.51-4.00 |
| 71-72 | EXECUTION TIME (MINUTES) FOR THE FIRST QUERY | 00-59 |
| 74-75 | EXECUTION TIME (SECONDS) FOR THE FIRST QUERY | 00-59 |
| 77-78 | EXECUTION TIME (1/100 SECONDS) FOR THE SECOND QUERY 1 | 00-99 |
| 83-84 | EXECUTION TIME (MINUTES) FOR THE SECOND QUERY | 00-59 |
| 86-87 | EXECUTION TIME (SECONDS) FOR THE SECOND QUERY | 00-59 |
| 89-90 | EXECUTION TIME (1/100 SECONDS) FOR THE SECOND QUERY | 00-99 |
| 92 | NORMAL FORM USED | 3 = THIRD<br>4 = FOURTH |

| | | |
|---|---|---|
| | QUERY NUMBER OF THE FIRST QUERY | 1 = QUERY 1<br>3 = QUERY 2<br>4 = QUERY 3<br>5 = QUERY 4 |
| 98-99 | NUMBER OF RELATIONAL OPERATIONS USED DURING THE FIRST QUERY | 00-99 |
| 101 | QUERY NUMBER OF THE SECOND QUERY | 1 = QUERY 1<br>3 = QUERY 2<br>4 = QUERY 3<br>5 = QUERY 4 |
| 105-106 | NUMBER OF RELATIONAL OPERATIONS USED DURING THE SECOND QUERY | 00-99 |
| 108 | COGNITIVE STYLE 1 * | 1 = EXTROVERTED<br>2 = INTROVERTED |
| 119 | COGNITIVE STYLE 2 * | 1 = FEELING<br>2 = THINKING |
| 110 | COGNITIVE STYLE 3 -- NOT USED * | |
| 111 | COGNITIVE STYLE 4 -- NOT USED * | |

* These values were provided by Ms. Sobel, Southern Methodist University, based upon analysis of the responses to the questions listed above.

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890
003        32225331112121254122121112211211221211121211222153224222300
12345678901234567890123456789012345678901
02 26 32 00 05 12 01 4 4 1 01 3 1 05 1221
005        11125421111121255121212211112121221111112211221253133333300
05 12 20 00 10 05 44 3 1 1 02 5 1 04 1122
006        12225321212111254222212211221221211221212112221154245234300
03 39 81 00 08 33 28 3 4 1 02 3 1 06 1112
007        11115422222222255122122112111122222221122211222154125333300
08 28 06 00 06 20 41 3 5 1 04 1 1 03 1221
008        11115512222222255 22.21221222121221112211222112254123233200
03 36 85 00 05 44 22 4 4 1 01 3 1 06 2211
009        12225242212222244111212221112222121222211121121254124114400
04 44 57 00 08 15 26 3 1 1 02 5 1 05 2112
012        12225232222222154212212221211 2.21212222212221222541333343 00
08 28 06 00 06 20 41 3 5 1 04 1 1 03 2111
013        12225242212222244111212221112222121222211121121254124114400
05 45 26 00 07 27 20 3 2 1 02 5 1 04 2112
014        12225532111111253222222221211221211211111211221154254224100
05 45 26 00 07 27 20 3 2 1 02 5 1 04 1112
015        11225521112222552121111222212122112222122222111153125324400
02 01 77 00 05 51 96 4 1 1 02 5 1 05 2211
016        21125521112122255222111212221 2.12111222112222211551543333 00
06 42 49 00 10 58 56 4 1 1 06 5 1 07 1211
017        21222521112121225121122112212112221112221222121113123223200
06 42 49 00 10 58 56 4 1 1 06 5 1 07 1221
018        32221331111121211111111112211112211212121211222111122143300
04 57 31 00 17 18 28 4 4 1 02 3 1 12 2221
019        22225431111111254122121212121212222121221121122215412211110
07 17 07 00 11 57 76 3 4 1 03 5 1 04 2211
020        12115521111222155212112122111111111222211221112153142321100
03 08 45 00 07 18 97 4 1 1 03 5 1 03 2212
021        11225512222222155122111122221112221212122212212215512533440
03 08 45 00 07 18 97 4 1 1 03 5 1 03 2221
022        12225242111122255111211212212252221211221222221254123333300
05 16 15 00 07 07 38 3 1 1 02 5 1 03 2111
023        22225431111111254122121212121212222121221121122215412211110
07 17 07 00 11 57 76 3 4 1 03 5 1 04 1222
024        12225522222222255222211121221222221212212212222215512333330
09 50 17 00 09 43 15 3 1 1 05 5 1 04 2211
025        12225232222221551 1.2122111211.22211222222122222551532212 00
12 25 34 00 12 40 94 3 5 1 06 1 1 05 2222
026        11125511111111251122212112211121221221111221222154233335300
08 35 14 00 17 48 13 4 1 1 02 5 1 07 1221
027        11125522112121255122222212112 2.22212211112121211541232354 00
02 32 09 00 14 17 01 4 1 1 02 5 1 07 1122
028        12125331112222155212211212121222211121122211112154153432200
05 53 98 00 08 54 31 3 5 1 05 3 1 08 2222
029        11225511111111155221221212221111211212222221222155154334400
15 01 22 00 09 17 77 3 4 1 06 3 1 08 1221
030        12215522111111155122212111112221221122111222122155134332200
04 13 81 00 15 26 15 3 4 1 02 3 1 08 1211
031        12215522111111155122212111112221221122111222122155134332200
```

04 13 81 00 15 26 15 3 4 1 02 3 1 08 1211
032 22214331112121155111111221121122221212112211222552452334 00
00 55 75 00 08 33 55 4 4 1 01 3 1 05 1122
033 12125531111222155121122212211111211211212212212155254 1243 00
06 01 63 00 09 31 72 4 4 1 01 3 1 06 1222
035 11125522112121255122222212122.22212211112121211541232354 00
02 32 09 00 14 17 01 4 1 1 02 5 1 07 1122
036 21125511112222155112112222221212211212211222112155155 1244 00
03 36 85 00 05 44 22 4 4 1 01 3 1 06 2212
037 22214331112121155111111221121122221212112211222552452334 00
05 53 98 00 08 54 31 3 5 1 05 3 1 05 2222
038 11225512222222155122111122221112221212122212212215512 53344 00
14 08 99 00 08 07 29 4 4 1 04 3 1 06 2221
040 11125521112121255222112112221212221212112211112153155 2222 00
05 53 98 00 08 54 31 3 5 1 05 3 1 05 2222
041 11124521112121221211112121112122221112112211121143254 3343 00
05 55 15 00 16 16 03 3 4 1 03 3 1 10 2122
042 11225511111111155221221212221111211212222221222155154 3344 00
15 01 22 00 09 17 77 3 4 1 06 3 1 08 1211
043 32221532222222212121211211222122112212121122121121542 42 00
02 52 03 00 17 29 79 4 1 1 02 3 1 10 1122
045 32211432112121213112222121112222221222112112122241215 2222 00
03 13 14 00 16 11 90 4 1 1 03 3 1 11 2122
046 11225242111111254222222211211221221221211211221541233333 00
04 55 34 00 08 16 53 4 4 1 02 5 1 03 1112
047 32221242222222215122212112111222221122122111221232254 4442 00
02 56 54 00 11 38 87 4 1 1 02 3 1 06 1122
048 312215311122.121312122221112212222222211111212222212333 33 00
02 52 03 00 17 29 79 4 1 1 02 3 1 10 2122
049 21214231111111133221222211212121221121221122221214143 4433 00
19 53 43 00 15 59 38 4 1 1 05 5 1 08 1112
050 31221431111111253121212221112222221222111111122254134 3343 00
01 55 90 00 06 12 01 4 1 1 02 5 1 04 2122
051 12213522112121234121122112211121221212122222221122123 3222 00
04 39 08 00 05 40 10 4 1 2 02 3 2 04 1222
052 31221531111111212112212111112212222122211211222111111 34233 00
01 43 90 00 19 35 53 4 4 1 01 3 1 11 2122
053 32221331212121223111212121212112222122211121112121225 42222 00
03 41 57 00 10 17 20 4 4 1 01 3 1 07 2122
054 32221331112121213121112212122222121222211122121141213 3333 00
02 59 90 00 06 57 16 4 4 1 01 3 1 06 2212
055 23222222111111223121221111212111221211122111221243223 3333 00
02 59 57 00 08 06 53 4 1 1 02 5 1 04 1122
056 31221512111111213121121111112111221112121212221141224 3221 00
03 34 17 00 17 27 43 4 1 1 02 5 1 13 1122
057 21125421111121134222222211112211221211111112212531213 113 00
01 32 11 00 05 58 66 4 1 1 01 5 1 03 1122
058 32221232212121211121122112112121211122111211222111225 2222 00
04 39 08 00 05 40 10 4 1 1 02 5 1 04 1222
059 21123511212121244122222212211111221221211112121252123 4244 00
08 35 65 00 05 35 38 4 1 1 04 5 1 05 1122

```
060        212245211121212211212112122112122211212111212222412542222 00
08 06 42 00 18 07 03 4 1 1 02 5 1 11 1112
061        222224211121212251212221111211212111211121112211122243332 00
12 28 08 00 09 16 82 4 1 1 04 3 1 07 1222
062        312214311111112231212122211122222212221111111222111143343 00
01 55 90 00 06 12 01 4 1 1 02 5 1 04 2122
063        322215212121212131112121211121222212221111111212412152233 00
03 01 31 00 04 27 04 4 1 1 02 5 1 04 2122
064        112245211121212441221221122221212111111222112221532222222 00
04 24 56 00 09 32 16 4 4 1 03 5 1 08 1221
065        211255111111112122212212112222212212112112221212121254451 00
15 20 00 00 07 23 66 3 1 1 12 5 1 06 1112
066        332211311121212141222221121112212112112112112211422552222 00
09 11 34 00 12 07 66 3 1 1 06 5 1 06 1222
067        211255311111124412222212112111112111121111121241332333 00
05 40 32 00 13 02 41 3 1 1 03 5 1 07 1221
068        222122312111112332111122111222222212222122211222241242222 00
06 41 07 00 10 57 68 3 1 1 04 5 1 11 2212
069        122253211121212131211212121111211222222222222222411243232 00
04 39 74 00 06 50 62 3 1 1 02 5 1 09 1212
070        222243222222222531211221221211122111221222112121544443344 00
05 02 09 00 10 01 92 3 1 1 02 5 1 04 1222
071        222232422122222241111212221112222121222211121121224124114 4 00
04 44 57 00 08 15 26 3 1 1 02 5 1 05 2112
072        321212422222222111221221112212212112111211111222452555555 00
17 26 13 00 32 54 57 3 4 1 06 5 1 28 1221
073        232222221111112231212211112121112212111221112212432233333 00
02 59 57 00 08 06 53 4 1 1 02 5 1 04 1121
074        121155211112221552121121221111111112222112211121531423211 00
03 08 45 00 07 18 97 4 1 1 03 5 1 03 2212
075        321212422222222111221221112212212112111211111222452555555 00
17 26 13 00 32 54 57 3 4 1 10 5 1 27 1221
076        111255211121212552221121122212122212121122111121531552222 00
04 55 34 00 08 16 53 4 4 1 02 5 1 03 2221
077        311115311111111243122122211121221221121111122221121213334 2 00
02 57 13 00 07 44 45 3 1 1 02 5 1 03 1222
078        322212411121112121122122112222.222221222211212222112322 32 00
06 50 30 00 08 21 41 3 1 1 04 5 1 03 2111
079        222232322222222251112122211212122122222122121221442542344 00
08 16 57 00 17 57 97 3 4 1 02 5 1 04 2222
080        311115311111111243122122211121221221121111122222115323333 42 00
02 57 13 00 08 06 53 3 1 1 02 5 1 03 1112
081        222243222222222231211221221211122111221222112121242243344 00
05 02 09 00 10 01 92 3 1 1 02 5 1 04 1222
082        222222312121212132112121211212222212222111211212222142242 00
05 00 93 00 08 49 92 3 1 1 02 5 1 03 2112
083        212142311111111332212222112121212211212211222212141434433 00
19 53 43 00 15 59 38 4 1 1 05 5 1 08 1112
084        222254311121112241212221111121221212221211122121441242243 00
05 24 88 00 14 26 67 3 1 1 03 5 1 07 1122
085        322212411121212151122221222112122222221211111221421323355 00
```

08 18 55 00 08 11 47 3 4 1 02 5 1 11 2221
086 1112552112222123312122212212212122111211111112124425333333 00
01 12 33 00 20 38 95 4 4 1 01 3 1 08 1122
087 3121152221212121312121211111122211122122211122113224233 2 00
02 33 02 00 11 51 78 4 4 1 03 3 1 04 2222
088 2212343222222223412112111221212122112121212122114425432 21 00
01 17 59 00 16 02 45 4 4 1 01 3 4 12 1122
090 2222543222222222412121222211122222122211121222113225323 1 00
03 02 19 00 05 37 30 4 4 1 02 3 1 06 2222
091 2322214222212121212222211111121122122111121122114254213 1 00
02 35 44 00 06 12 56 4 4 1 01 3 1 04 1122
092 1112552211111122412222211222212122122111121122224312341 51 00
01 32 15 00 05 59 89 4 4 1 01 3 1 04 1222
093 3222143222222221311121212112212222 1222.11111121241214 4444 00
07 02 65 00 13 41 35 4 4 1 01 3 1 07 2122
095 2222443211111122322222221221221211211111211221114254224 1 00
01 32 11 00 05 58 64 4 4 1 01 3 1 04 1222
096 2321312222222222511121212111222212122222112112124415312 42 00
12 25 34 00 12 40 94 3 5 1 06 1 1 05 2112
097 3222123222222221222222122221222222121221111222122213555 55 00
04 21 89 00 19 48 81 4 4 1 01 3 1 10 2111
098 3222133111112121111111111221111221121212121122221111221433 00
04 57 31 00 17 18 28 4 4 1 02 3 1 10 2221
099 2112552111111122312112222121222212112211111212211242234455 00
19 35 67 00 23 29 39 3 4 1 12 5 1 17 1222
100 2222233111212121412212111221121122121112121122211322422 23 00
02 26 32 00 10 01 92 4 4 1 01 3 1 05 1221
101 2322512211111124322121221112221121112221112212122413333 33 00
07 31 54 00 14 53 35 4 1 1 09 3 1 18 1112
102 3221133111212122122221211121221212122222122212122212321 32 00
01 06 13 00 07 57 90 4 1 1 01 3 1 07 2111
103 2122233211111122122122222212222121121222112222111212344 44 00
02 30 60 00 23 24 00 4 1 1 01 3 1 10 1212
134 3121153111211124412111111112222222122111211122112221322 22 00
03 08 19 00 05 39 12 4 4 1 02 3 1 06 2222
134 1112542111112125512121221111212122111111221122125313333 33 00
04 01 63 00 06 44 66 4 4 1 01 5 1 05 1122
135 2112242111112112122222221111221122112111111122121112131 13 00
01 32 11 00 05 58 56 4 1 1 01 5 1 03 1112
136 2222224211111122422222221121122122122121112112214412333 33 00
04 55 34 00 08 16 53 4 4 1 02 5 1 03 1112
137 3222133111112122221111111111112121212121212121212141223 1222 00
05 21 03 00 20 31 93 4 4 1 02 3 1 06 1222
138 3111153111111122212122211212212222122111121122111121422 22 00
04 36 06 00 09 08 72 4 4 1 03 5 1 07 1122
139 3222153211212121312121211122122122122211111112224111533 33 00
04 03 87 00 06 05 26 4 4 1 01 5 1 04 2122
140 3112151121212122412222221221111122122121111212124212342 44 00
08 35 65 00 05 35 38 4 1 1 04 5 1 05 1111
141 2122553121111122512212222211111122121122221112114313333 33 00
06 12 06 00 21 52 59 3 4 1 04 5 1 10 1221

```
142       3211152211111121311212111212112121122112221122212221431111 00
11 20 67 00 08 30 27 3 4 1 04 5 1 05 1221
104       3122143111111123211121212112111221122211221122212442233333 00
13 02 30 00 17 28 50 3 1 1 03 3 1 09 2222
105       2112323111212122512122221211222122112111121112112422443333 00
05 55 15 00 16 16 03 3 4 1 03 3 1 11 1122
106       3322115222222221312112222121112212211211122112211232235555 00
08 18 55 00 11 30 14 3 1 1 03 3 1 06 1222
107       2212343222222223412112111221212122112121212212114425432221 00
01 17 59 00 16 02 45 4 4 1 01 3 1 13 1122
108       1222532111212121312112121211112112222222222222222411243232 00
04 39 74 00 06 50 62 3 1 1 02 5 1 09 1212
109       2112453111111123412121222221212122121211211121244154332 2 00
05 40 32 00 18 06 86 3 1 1 03 3 1 12 2122
110       3122133111212121122122222212222122122121122222211115324 42 00
13 30 48 00 41 43 45 3 4 1 02 3 1 24 1212
112       3222124221222121312222212221221222222111222212122222333 33 00
08 40 91 00 07 14 13 3 4 1 04 3 1 09 2121
113       2212353111111132511121212222212222122211211121244154222 2 00
05 26 37 00 16 54 42 3 4 1 03 3 1 10 2122
114       2222343111212123512121211121222122212222122211211241244333 00
06 04 48 00 12 42 75 3 4 1 03 3 1 10 1122
115       3122123221212112322222222222121112211212212212241122122 00
10 58 50 00 16 27 07 3 4 1 05 3 1 13 1111
116       2221553121212123512112222112122121121122111222212412234 42 00
14 11 41 00 33 35 38 3 4 1 06 3 1 15 1211
117       3322113121212121212112111111222121122111122222212322421 32 00
02 51 75 00 11 37 38 4 4 1 01 3 1 09 1222
118       1222552222222225522221112122122222121221221222215512333 33 00
08 16 57 00 17 57 97 3 4 1 02 5 1 04 2221
119       1211552111122215521211212211111111122221122111215314232 11 00
01 21 79 00 13 12 72 4 4 1 01 5 1 03 2212
120       3222154111212121322121222111211212122212212212224111433 33 00
03 49 65 00 07 25 50 4 4 1 01 5 1 04 2112
121       3222122111112122322121222112222222122121122112122212433 33 00
02 32 31 00 21 19 10 4 4 1 01 3 1 08 2112
124       3221153222222221212111212111222222122211211111212432244444 00
02 44 33 00 06 35 03 4 4 1 01 3 1 08 2122
125       3111152121212122312222111221222122122112111122212111422 22 00
02 39 51 00 16 14 32 4 4 1 01 3 1 10 1121
127       3122152111212122212212211222212121111112221122211112222 22 00
04 23 45 00 12 21 22 4 4 1 03 5 1 07 1221
128       3122133221112121212222211111121212212211221112211412133241 00
03 09 17 00 13 34 65 4 4 1 01 3 1 08 1122
129       2222222221122122221212212221222212221222122212222112112 22 00
11 41 52 00 20 21 43 3 4 1 02 5 1 14 2211
130       3122123221112121212212211111212121111111111222214121522 31 00
03 29 81 00 11 02 95 4 4 1 02 5 1 05 1121
131       3322114211111121211121211111212221122212121122214121353 54 00
04 18 43 00 08 59 64 3 4 1 03 5 1 05 2122
132       3221124111111121412112221222212111121111222211113152225 3 00
```

06 55 56 00 19 39 47 4 4 1 02 5 1 08 1212
133 22225332111111235121222221222212121212211112224322432222 00
01 21 79 00 13 12 72 4 4 1 01 5 1 03 2121
143 32221232212222211122112111122112222122211221112124122422224 00
13 09 66 00 23 30 17 3 4 1 04 5 1 15 2122
145 12215522111111155122212111112221221122111222122155134332 2 00
04 13 81 00 15 26 15 3 4 1 02 3 1 09 1112
146 31221431111111253121212221112222221222111111122254134334 3 00
01 55 90 00 06 12 01 4 1 1 02 5 1 04 2122
147 22224322222222253121122122121112211122122211212154444334 4 00
05 02 09 00 10 01 92 3 1 1 02 5 1 04 1222
148 22224322222222253121122122121112211122122211212154444334 4 00
05 02 09 00 10 01 92 3 1 1 02 5 1 04 1222
149 31111531111111243122122211121221221121111122221153233334 2 00
02 57 13 00 10 02 03 3 1 1 02 5 1 03 1112
150 31221431111111253121212221112222221222111111122254134334 3 00
01 55 90 00 06 12 01 4 1 1 02 5 1 04 2122
151 12215522111111155122212111112221221122111222122155134332 2 00
04 13 81 00 15 26 15 3 4 1 02 3 1 08 1112
153 32221242212221213122222122212212222221112222212122223333 3 00
05 02 93 00 09 08 98 3 4 1 01 3 1 06 2121
154 31111521112121222122121212222212211112122111122121113454 4 00
04 25 84 00 00 00 00 3 1 1 02 3 2 00 1221
155 32221242222222215122212112111222221122122111221232225444 2 00
02 56 54 00 11 38 87 4 1 1 02 3 1 06 1122
156 32221331112121213121112212122222121222211122121141213333 3 00
02 59 96 00 07 04 16 4 4 1 01 3 1 06 2112
157 32221241112121251122221222112122222221211111221421323355 00
05 02 93 00 08 52 98 3 1 1 02 5 1 03 2121
158 12225242111122255111211212212252221211221222221254123333 3 00
05 17 85 00 07 32 38 3 1 1 02 5 1 03 1111
161 11225521112222255212111122221212211222122222111153125323 3 00
02 00 08 00 05 51 96 3 5 1 06 1 1 05 2211
163 12225522222222255222211121221222221212212212222215512333 33 00
09 50 17 00 10 48 99 3 1 1 05 5 1 04 2211
164 22215531212121235121122221121221121121122111222212412234 42 00
14 22 41 00 33 35 38 3 4 1 06 3 1 15 1212
165 11125421111121255121212211112121221111112211221253133333 3 00
03 49 81 00 08 43 28 3 1 1 02 5 1 04 1122
167 21125421112222225121222212212221121112211121121124123445 5 00
19 09 76 00 28 06 65 3 4 1 07 5 1 12 1112
170 22222231112121223122212121112212121221211122122222124111 1 00
20 10 07 00 23 04 81 3 4 1 07 5 1 06 2112
171 12225432111111235121222221222212121212211112225322432222 00
00 58 41 00 17 16 45 4 4 1 02 5 2 00 2121
172 11225431111111253111211222111112121112221122122254123222 1 00
09 09 64 00 20 22 32 3 4 1 03 5 1 04 2111
173 22224521211111255222212222211222221222211222221124125333 3 00
06 01 63 00 09 31 72 4 4 1 01 5 1 05 2112
174 32222241212121223121222111111211221121211122111214153322 2 00
09 09 53 00 12 09 68 3 4 1 02 5 1 04 1112

```
175        22225431111111233111211222111112121112221122122224123 2221 00
09 10 69 00 20 28 32 3 4 1 03 5 1 04 2111
176        22225541111111235212222121221212221212211122221124123 2211 00
28 06 43 00 20 27 93 3 4 1 11 5 1 12 2111
177        31221521112121135122111112112121221221121212222121222232 2 00
04 58 30 00 11 17 29 3 4 1 03 5 1 06 1121
178        32221231112121213121122112121121211111122112222141223 2222 00
06 29 32 00 08 24 38 3 4 1 03 5 1 04 1222
179        22225432222222224121212222111222221222111212221113225323 1 00
02 35 44 00 06 12 56 4 4 1 01 3 1 06 2112
180        21224521111111145222221112211111211211122221211223223 2333 00
05 24 94 00 11 07 89 3 4 1 02 5 1 09 1121
181        22222231111111233122211112112111221212211121121244233113 2 00
06 02 84 00 17 40 89 3 4 1 02 5 1 05 2121
182        32221241112121212212211221221212121222121111121242123231 2 00
08 40 91 00 07 14 13 3 4 1 04 3 1 09 2111
183        22222332212121133122222211122221211121222111222243124414 1 00
09 24 56 00 06 54 45 4 4 2 04 5 2 04 1122
184        11225511212111255221222111112221221111221222222124134323 2 00
05 34 82 00 11 21 89 3 4 1 03 5 1 05 1112
185        22222221211121225122222211111111221121211211221224122223 3 00
15 35 45 00 13 43 21 3 4 1 08 5 1 09 1122
186        32221232212222223112221122212222211221122111221122122244 00
11 59 64 00 12 04 25 3 4 1 03 5 1 11 2121
187        22223331122121234122212111212122221211111212121112123333 3 00
02 55 87 00 08 36 74 3 4 1 05 5 1 09 2121
188        32221241112121213222221211112221112122121211121124222333 3 00
05 01 16 00 26 30 15 3 4 1 03 5 1 07 1112
189        33211211112121215121122111212111211122112211111143155233 3 00
10 35 66 00 05 35 38 4 4 1 05 5 1 04 1122
190        33221251112222211222222112122221121211221122121222325544 00
09 54 68 00 21 00 95 3 4 1 02 5 1 09 1111
192        21224521111111455222221112211111211211222212112532232333 00
08 31 23 00 09 23 45 4 4 1 04 5 1 06 1121
193        11224541212121243121222111111211221121211122111254153322 2 00
09 09 53 00 12 09 68 3 4 1 02 5 1 04 1112
194        22224521211111255222212222211222221222211222221124125333 3 00
06 01 63 00 09 31 72 4 4 1 01 5 1 05 2112
195        12224242222121243122222111111211221221111211221154254213 1 00
10 23 23 00 07 29 34 4 4 1 04 5 1 03 1122
196        21125421112222225121222122122211211122111211211241234455 00
10 03 81 00 27 01 30 3 4 1 07 5 1 12 1112
197        11225521112222255212111222212122112221222221111531253233 00
02 01 77 00 05 51 96 4 1 1 02 5 1 05 2211
198        12225242111122255111211212212252221211221222221254123333 3 00
06 06 15 00 09 34 38 3 1 1 02 5 1 03 1111
199        32221241112121215112222122211212222222121111122142132355 00
05 10 93 00 09 19 92 3 1 4 02 5 1 03 2121
201        21225521112121253221212221112112121222121222122251114333 3 00
03 49 65 00 07 25 50 4 4 1 01 5 1 04 2112
202        31211531112111244121111111222222212211121112211222132222 00
```

04 04 63 00 06 58 66 4 4 1 01 5 1 05 1122
203 211255211111112231211222121222212112211111212211242234455 00
18 03 43 00 29 32 23 3 4 1 12 5 1 17 1212
204 112252322111212531221221111121212111111111122221532152231 00
03 29 81 00 11 02 95 4 4 1 02 5 1 05 1122
205 121253311122221552122112121212222111211222111121541534322 00
05 58 98 00 10 54 31 3 5 1 05 3 1 05 2222
208 332211421111112121112121111121222112221212112221412135354 00
04 58 03 00 10 09 88 3 4 1 03 5 1 05 2122
210 111255111111111551112122222122221222122111221212541542243 00
06 46 78 00 06 36 07 3 1 1 05 3 1 06 2111
211 211255111111112551211221122211212112111212111221552441243 00
04 59 84 00 06 44 74 3 1 1 02 3 1 05 1221
212 212144211111112442221212222112212212112112222121124123234 2 00
04 59 84 00 07 52 76 4 1 1 02 3 1 06 1211
213 211255111111112551211221122211212112111212111221551541442 00
10 08 03 00 06 44 74 3 5 1 07 3 1 05 1222
214 212155211111112542221212222112212212112112222121154123234 2 00
04 59 84 00 07 52 76 4 1 1 02 3 1 06 1211
215 211255111111112551211221122211212112111212111221551541442 00
05 44 57 00 08 15 26 3 1 1 02 5 1 05 1221
216 222233422222222151222121112111222212211212121212551442343 00
01 47 82 00 04 04 53 4 1 1 02 3 1 05 2121
217 222233422222222151222121112111222212211212121212551442343 00
04 04 53 00 04 27 48 4 3 1 05 5 1 03 2121
218 222233422222222151222121112111222212211212121212551442343 00
04 04 53 00 04 27 48 4 3 1 04 5 1 03 2121
219 322212411121212132222212111122211212212121112112422233333 00
10 08 03 00 06 44 74 3 5 1 07 3 1 05 1112
220 212155211111112542221212222112212212112112222121154123234 2 00
02 52 75 00 04 04 80 4 3 1 05 5 1 03 1211
221 111255111111111551112122222122221222122111221212541542243 00
06 48 78 00 06 36 07 3 1 1 05 3 1 06 2111
222 212144211111112442221212222112212212112112222121124123234 2 00
02 52 75 00 04 04 80 4 3 1 05 5 1 03 1211
223 211255111111112551211221122211212112111212111221551541244 00
03 58 95 00 05 52 23 4 4 1 02 5 1 03 1221

MONTANA STATE UNIVERSITY LIBRARIES
3 1762 10025525 4

AMERICAN BINDERIES