

GRADUATE STUDENT RECORDS RELATIONAL
DATA BASE DESIGN

by

JOHN L. COOK

B.S., Embry Riddle Aeronautical University, 1978

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

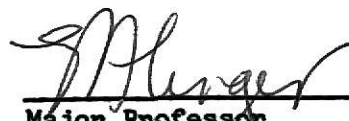
MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1982

Approved by:


Major Professor

SPEC
COLL
LD
2668
.R4
1982
C66
C.2

111203 652166

TABLE OF CONTENTS

	Page
LIST OF FIGURES	iii
ACKNOWLEDGMENTS	iv
CHAPTER	
I. INTRODUCTION	1
1.0 Background	1
1.1 Overview	3
1.2 Objectives	5
1.3 Report Structure	6
II. DATA BASE DESIGN	8
2.0 The Basic Data Base Design Challenge	8
2.1 Problems in Logical Data Base Design	9
2.2 A New Approach: The Entity-Relationship Model	10
2.3 Advantages of the Entity-Relationship Approach	10
2.4 Objective and Methodology of the Data Base Design	13
2.5 Requirements Formulation and Analysis	14
2.6 The Conceptual/Logical Design (E-R Diagrams)	15
2.6.1 Selection of Entities	16
2.6.2 Selection and Identification of Relationships	21
2.6.3 Selection of Entity Attributes and Value Types	38
2.6.4 Selection of Key Attributes for Entities	41
III. DATA BASE MODELS	42
3.0 Data Base Systems and Models	42
3.1 The Relational Model	42
3.1.1 Advantages of the Relational Approach	44
3.1.2 Challenges of the Relational Approach	46
3.2 Normalization and Functional Dependency	47
3.3 Automatic Data Base Generator System	50
3.4 Relations	53
3.4.1 Initial Design	53
3.4.2 ADBG Design I	53
3.4.3 Second Design	55
3.4.4 ADBG Design II	56
3.4.5 ADBG Commands	56
IV. GRADUATE RECORD DATA BASE DESIGN	58
4.0 Schema	58
4.1 Data Dictionary	60
4.2 Design Considerations	61

**THIS BOOK
CONTAINS
NUMEROUS PAGES
WITH DIAGRAMS
THAT ARE CROOKED
COMPARED TO THE
REST OF THE
INFORMATION ON
THE PAGE.**

**THIS IS AS
RECEIVED FROM
CUSTOMER.**

	ii
V. SUMMARY AND FUTURE ENHANCEMENTS	78
5.0 Future Enhancements	78
5.0.1 Design	78
5.0.2 Security	78
5.0.3 Data Dictionary	78
5.0.4 Consistency	79
5.1 Summary	79
BIBLIOGRAPHY	83
APPENDICES	
A INITIAL ENTITY DIAGRAMS WITH RELATIONSHIPS, ATTRIBUTES AND VALUES	A-1
B FINAL ENTITY DIAGRAMS WITH RELATIONSHIPS, ATTRIBUTES AND VALUES	B-1
C ADBG DESIGN I	C-1
D ADBG DESIGN II	D-1
E FINAL E-R DIAGRAM	E-1

LIST OF FIGURES

FIGURE		Page
1.1	Master Card	2
2.1	Enterprise Schema--An Intermediate Step in Logical Data Base Design	11
2.2	STUDENT_ADMITTANCE INFORMATION	24
2.3	STUDENT_ADVISOR	25
2.4	STUDENT_MAJOR PROFESSOR	26
2.5	STUDENT_ADVISORY COMMITTEE	27
2.6	STUDENT_UNIVERSITIES	28
2.7	STUDENT_STATUS	29
2.8	STUDENT_MASTERS RESEARCH	30
2.9	STUDENT_PROGRAM OF STUDY	31
2.10	STUDENT_TEACHING ASSISTANT	32
2.11	TEACHING ASSISTANT_COURSE LIST	33
2.12	MAJOR, SUPPORTING, DEFICIENCY COURSE_COURSE LIST	34
2.13	PROGRAM OF STUDY_MAJOR, SUPPORTING, DEFICIENCY COURSE	35
2.14	Initial GSRDB E-R Diagram	36
2.15	E-R Diagrammatic Technique Example	37
3.1	Bernstein's Algorithm #2	52
4.1	GSRDB Data Dictionary	62
A.1	Student Entity	A-2
A.2	STUDENT_ADVISOR Relationship	A-3
A.3	STUDENT_MAJOR PROFESSOR Relationship	A-4
A.4	STUDENT_ADVISORY COMMITTEE Relationship	A-5
A.5	STUDENT_STATUS Relationship	A-6
A.6	STUDENT_ADMITTANCE INFORMATION Relationship	A-7
A.7	STUDENT_TEACHING ASSISTANT Relationship	A-8
A.8	STUDENT_MASTERS RESEARCH Relationship	A-9
A.9	MAJOR, SUPPORTING, DEFICIENCY COURSE_PROGRAM OF STUDY Relationship	A-10
A.10	MAJOR, SUPPORTING, DEFICIENCY COURSE_COURSE LIST Relationship	A-11
A.11	TEACHING ASSISTANT_COURSE LIST Relationship	A-12
A.12	STUDENT_PROGRAM OF STUDY Relationship	A-13
A.13	STUDENT_UNIVERSITIES Relationship	A-14
B.1	Student Entity	B-2
B.2	STUDENT_GRAD COURSES	B-3
B.3	COURSE_LIST_GRAD COURSES Relationship	B-4
B.4	STUDENT_STATUS Relationship	B-5
B.5	STUDENT_ADMITTANCE INFO Relationship	B-6
B.6	STUDENT_TEACHING ASSISTANT Relationship	B-7
B.7	STUDENT_MASTERS RESEARCH Relationship	B-8
B.8	STUDENT_UNIVERSITIES Relationship	B-9
B.9	STUDENT_GPA Relationship	B-10
E.1	Final E-R Diagram	E-2

ACKNOWLEDGMENTS

I wish to express my sincere thanks to Dr. Elizabeth Unger, Professor in the Computer Science Department, Kansas State University. Dr. Unger provided invaluable guidance, direction and assistance in the preparation of this report.

I am indebted to Dr. Paul Fisher, Professor in the Computer Science Department, Kansas State University, for providing me the opportunity to work with the Automatic Data Base Generator system.

I am sincerely appreciative of the assistance provided to me by Kwan Chul Park, Ltc. in the Korean Army and Ph.D. candidate in the Computer Science Department, Kansas State University.

My sincere appreciation goes to Rhonda Terry who did a superb job in assisting me in editing and typing this report.

To my wife Roberta and to my son John Jude, were it not for their continual support, inspiration, and sacrifices this report would not have been written and my goal would not have been achieved. I am deeply indebted and thankful for the love and encouragement that they continually provided me.

CHAPTER I

INTRODUCTION

1.0 Background

Currently the Computer Science Department employs a manual system for processing graduate student records in the Department. The process consists of opening a master card for each applicant and then posting data to this card as processing requirements are met. Figure 1.1 shows a sample of a master card.

Upon opening a master card, an application folder for each student is also initiated. This folder is used to place all forms, records, and transcripts that are required by the Graduate School and the Computer Science Department. The master card is posted and forms and records are filed as requirements are met and documents are obtained. When all requirements are met, the file is sent to the Graduate Student Committee for review. Based on this review, an applicant is either recommended for admittance or nonadmittance. This recommendation is forwarded to the Graduate School and posted to the master card.

If the applicant has been recommended for admittance, his/her folder is moved to the accepted file and appropriate entries to the master card are made. However, if the applicant has not been favorably considered for admittance, his/her file is placed in the rejected area of the file and the master card is appropriately annotated. In either instance, an appropriate letter is sent to the applicant.

The accepted applicant's folder remains in the accepted area of the file pending enrollment to the requested semester. If the student does not enroll in that semester, his/her folder is moved to the inactive

NAME _____	LOCATION: KSU	KC	LEAV	NCR	WE
ADDRESS _____	LEVEL : MS	PhD	Special		
_____	STARTING: Fall	SP	SU	19__	
_____	STATUS : Missing letter sent				
_____	To Committee				
Need Assistance YES NO	To Xerox				
KSU FORM _____	To Grad. Sch.				
CS FORM _____	Status letter				
TRANSCRIPT (2 copies)	TOEFL _____ () HEALTH _____ ()				
_____ ()	\$8000 _____ ()				
_____ ()	Recommendation: ACCEPT REJECT				
_____ ()	PROBATION SPECIAL				
RECOMMENDATIONS	PROVISIONAL TA RA				
_____ ()	DATE RECEIVED _____				
_____ ()	EXPERIENCE _____				
_____ ()	_____				
Acceptance letter _____ ()	_____				
Rejection letter _____ ()	_____				

Figure 1.1
Master Card

location of the file, while the rejected applicants folder remains in the rejected area of the file for a maximum of five years.

During the time a student is participating in the Graduate Program, the folder is a repository for all forms and records accumulated during his enrollment. However, the master card is destroyed upon enrolling in the Computer Science Department.

This manual process, while providing a somewhat satisfactory means to track admission requirements, lacks visibility and easy access needed to review or update an applicant's status without laboriously reviewing both the master card and each form in every applicant's folder. Additionally, after the applicant is accepted, student status and progress reviews cannot be made in a timely fashion because of a lack of readily available pertinent information, necessitating the same laborious review of each folder.

An automated system that is manageable, efficient, provides visibility, easy access and updating and is designed to store and maintain all relevant data items concerning both the applicant and student records from the initial application stage until graduation is needed. This would provide the Department with ability to perform its function in a timely manner and better serve the student.

1.1 Overview

This report describes the methods used and resultant design for a Computer Science Graduate Student Records Data Base (GSRDB). The impetus for this project is the Computer Science Department's need for an efficient, user friendly, responsive, and manageable automated system

that provides pertinent information to perform all applicant/student related administrative tasks. The data base that I have designed will provide the Department with the relations that are necessary to store and maintain a collection of all relevant data items concerning applicants and enrolled students. This data base, when coupled with a user friendly transaction processor and output processor, will provide the information necessary to perform all administrative functions and provide the ability to monitor student progress in a timely, efficient manner. The INGRES DBMS has been selected for implementation of this data base design.

INGRES is a relational DBMS designed to provide users with easily understood queries. The data manipulation language supported by the INGRES system is called QUEL (QUERy Language). INGRES was selected for two basic reasons. First, it is currently available at the Computer Science Department and, secondly, relational data bases have the following attractive features:

1. simplicity and ease of use,
2. data independence,
3. symmetry,
4. flexibility, and
5. good theoretical foundation.

These features are discussed in detail in Section 3.1.1, Advantages of the Relational Approach.

1.2 Objectives

The objectives of this research are:

1. To gain practical experience in data base design.
2. To study an integration of design methodologies consisting of design techniques, tools, and analysis that can be used to arrive at the data base design that accommodates the application functions of a particular organization.
3. To become knowledgeable of the Entity-Relationship (E-R) model, use it, and determine its benefit in the data base design process and to create an E-R diagram for use in long term data base planning within the Department.
4. To gain practical experience and knowledge concerning the coordination required to achieve a successfully integrated software project.
5. To design a data base for implementation on a Relational DBMS.
6. To design relations to support a user friendly interactive system that minimizes operator input requirements.
7. To design the relations so that all data can be stored while minimizing the number of relations in order to reduce the processing and programming required to add and update records and produce standard output reports.
8. To provide the Computer Science Department with an efficient, relational data base design that supports a user friendly

transaction processing system, which will enable the Department to automate graduate student administration and monitoring.

9. To assist the Computer Science Department's efforts to provide better service for graduate applicants/students while reducing the staff and faculty manhours required to perform student administrative advising and monitoring functions.

1.3 Report Organization

Chapter II is concerned with data base design. Its main focus is how to arrive at a logical/conceptual view of the data base. The Entity-Relationship (E-R) model is the vehicle that was selected to accomplish this task. The construction of the E-R model and diagrammatic technique for representing entities, relationships, attributes and value types is explained. The chapter concludes with a discussion on the selection of key attributes.

Chapter III begins by discussing and comparing the hierarchic, network, and relational data base models. Since the relational model was selected for this data base implementation, a discussion concerning the advantages and challenges of the relational model is presented. The concerns of normalization and functional dependencies are of great importance in a relational data base; therefore, a fairly comprehensive review of these topics is presented. This project incorporated the use of the Automatic Data Base Generator (ADBG) system to provide a data dictionary [9], check functional dependencies, and present a third

normal form (3NF) schema. The chapter concludes with the results and implications of using this system.

Chapter IV presents the schema for the Graduate Student Records Data Base design. The rationale and design considerations for this schema are discussed. The chapter concludes after presenting the data dictionary developed for the GSRDB.

Chapter V discusses future enhancements to the system and concludes with a summary of the design.

CHAPTER II

DATA BASE DESIGN

2.0 The Basic Data Base Design Challenge

There are no industry-wide standards for systems analysis tools, forms or languages for data base design. Organizations and information system designers employ various systems analysis methods. In an effort to systematize the information system development process, more and more of the larger organizations are establishing some kind of systems analysis standards [2]. However, a challenge to both researchers and practitioners is to devise better systems analysis practices and tools (perhaps a futuristic computer language and processor for system analysis through which the computer itself may assist in checking for the consistency, completeness, and so forth of the information system) [2].

A major aim of the initial systems analysis effort is to arrive at a conceptualization of the data base, independent of the hardware and data model. This conceptualization should not be a particular user's view but rather a global view of the entire data base. The formulation of this view is an intermediate step in arriving at the logical data base design. This intermediate step in logical data base design is needed because the logical data base design is the basic challenge: it is the data base design's representation of the real world. Successfully meeting this challenge increases the likelihood that the data base being designed will meet the user (organization) requirements.

2.1 Problems in Logical Data Base Design

Data base design today is a complicated process. The designer has to consider not only how to model the real world, but also the limitations of the data base system and the efficiency of retrieval and updating processes. Examples are:

1. The data base designer is constrained by the limited data structure types supported by the data base system. For example, the many-to-many relationships between two types of entities, such as the relationships between faculty and students cannot be represented directly in many data base systems [5].
2. The data base designer may have to consider the access path of the records (e.g., how to access a particular record type) [5].
3. The data base designer may want to consider how to make the retrieval and updating more efficient. Thus, the data about an entity in the real world may be put into more than one record for efficiency purposes. For instance, the data items about a student may be grouped into two records: STUDENT_MASTER and STUDENT_DETAIL [5].

The problems listed above reinforce the fact that the data base designer has to consider many issues at the same time, which makes the data base design task very difficult. Additionally, the final output of the logical data base design process is the user schema (e.g., a description of the user view of the data base). Since the user schema

represents the data base designer's solution to the complicated issues mentioned above, it is not difficult to see why user schemata are usually difficult to understand and difficult to change.

2.2 A New Approach: The Entity-Relationship Model

This approach to logical data base design is called the Entity-Relationship (E-R) approach. The key idea of the E-R approach is to add an intermediate stage in logical data base design (see Figure 2.1). The data base designer first identifies the entities and relationships which are of interest to the enterprise using the Entity-Relationship (E-R) diagrammatic technique. At this stage, the data base designer should view the data from the point of view of the whole enterprise (not the view of a particular user). Therefore, we shall call the description of the enterprise view of data the "enterprise schema" [5]. The enterprise schema should be a "pure" representation of the real world and should be independent of storage and efficiency considerations. The data base designer first designs the enterprise schema and then translates it to a user schema for his data base system (see Figure 2.1).

2.3 Advantages of the Entity-Relationship Approach

Conventional approaches to logical data base design usually have only one phase: mapping the information about objects in the real world directly to a user schema [5]. The E-R approach to logical data base design consists of two major phases: (1) defining the enterprise schema

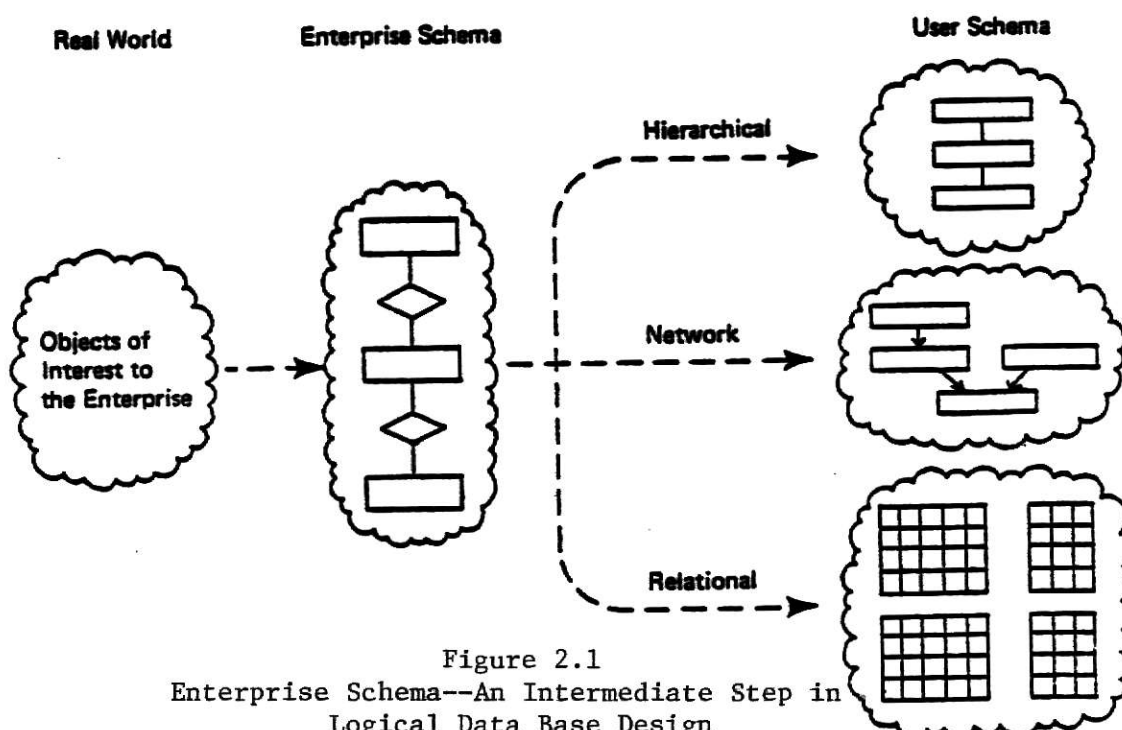
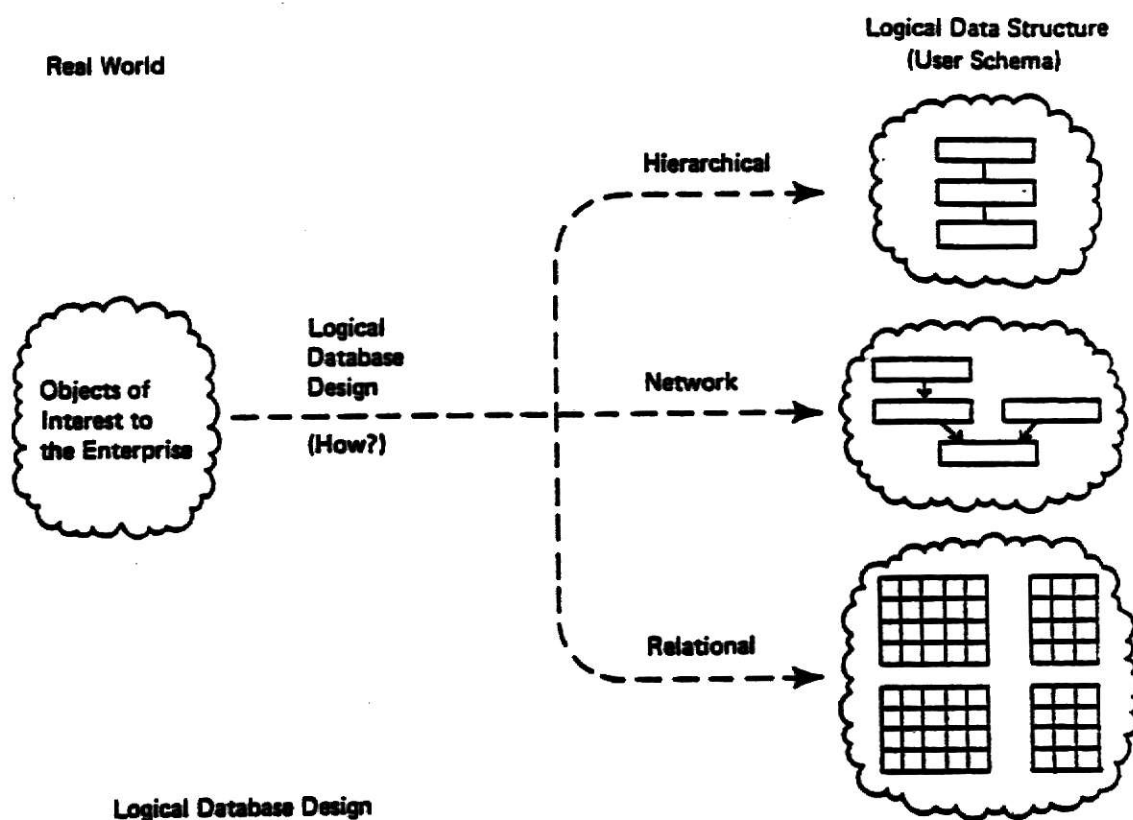


Figure 2.1
Enterprise Schema--An Intermediate Step in
Logical Data Base Design

using the Entity-Relationship diagram, and (2) translating the enterprise schema into a user schema. The advantages of using the E-R approach are:

1. The division of functionalities and labor into two phases makes the data base design process simpler and better organized [5].
2. The enterprise schema is easier to design than the user schema since it need not be restricted by the capabilities of the data base system, and is independent of the storage and efficiency considerations [5].
3. The enterprise schema is more stable than the user schema. If one wants to change from one data base system to another, one would probably have to change the user schema but not the enterprise schema, since the enterprise schema is independent of the data base system used. What needs to be done is to remap the enterprise schema to a user schema suitable for the new data base system. Similarly, if one wants to change the user schema to optimize a new application program, one need not change the enterprise schema, but rather remap the enterprise schema to a new user schema [5].
4. The enterprise schema expressed by the Entity-Relationship diagram is more easily understood by non-EDP people and serves as an excellent medium for communication between the designer and management personnel.

5. Constructing an E-R diagram allows the designer to concentrate solely on the entities and their relationships. This concentrated focus on the front-end of the design enables the designer to become intimately associated with the entities and their relationships and should provide a more logical design that can be more effectively implemented.
6. Producing the E-R diagram adds documentation to the design process which should be beneficial for any future enhancement of the system.

2.4 Objective and Methodology of the Data Base Design

The basic objectives of data base design are to enable users to obtain the exact data they need to perform their duties within the organization and to make the data available in a reasonable amount of time. The first objective requires that the elements in the data base represent the complete data needs of the user organization based on their overall goals, internal organizational structure, and projected data access requirements. The second objective, performance, requires that the data base structure resulting from the design process allow fast enough access to data to allow those who need the data to perform their duties effectively. Performance has present and future aspects, however [11]. The early design-phase goal could be to produce a flexible data base structure that should be adaptable to a changing user environment, whereas the later design phase may emphasize tuning to optimize performance for known processing requirements [11].

The major objective of this design is to enable the Computer Science Department users to access a data base that contains all the data needed to perform student records administration, processing, and monitoring in an extremely efficient manner.

In order to accomplish this task, an integrated stepwise data base design methodology incorporating E-R diagrams in step 2 (Conceptual/Logical Design) and the Automatic Data Base Generator in step 3 (Conceptual/Logical Design) was used.

This practical methodology follows design steps that represent the major principles developed in currently known data base design techniques. The steps in this methodology are:

1. Requirements Formulation and Analysis
2. Conceptual/Logical Design (E-R Diagrams)
3. Conceptual/Logical Design (Normalized Relations via the Automatic Data Base Generator system)
4. Implementation Design (translation from Normalized Relations to the data base schema)
5. Physical Design

The first step in this methodology is the requirements formulation and analysis process; it is presented in the next section.

2.5 Requirements Formulation and Analysis

The requirements formulation and analysis was the first activity performed in designing this data base. This activity involved the establishment of organizational objectives, derivation of specific data base requirements from those objectives or directly from management and nonmanagement personnel, and documentation of those requirements in a form that was agreeable to the users and the designer. The technique used was to interview the users, management and key employees involved

in the administration of student records. Then all documents associated with student administration and records were collected and analyzed. At the completion of the analysis, 29 major data elements had been identified. The results of this analysis and the user interviews were studied and documented. A meeting was then conducted with management and key personnel. The purpose of this meeting was to discuss the results of the document analysis and interviews. The meeting was successful and confirmed that the flow of data elements associated with the administration process, interfaces between the processes, and that management, user, and the designer correctly understood the flow and semantics of the data elements and their potential values. Additionally, issues regarding security and the freedom of information act were discussed. Essentially, the requirements formulations and analysis phase produced the major data element names, data element values and data element usage. With this information agreed upon, the design proceeded to step 2, the conceptual/logical design.

2.6 The Conceptual/Logical Design (E-R Diagrams)

The first stage, conceptual design phase, concerns itself with the description and synthesis of diverse users' information requirements into a preliminary data base design [11]. This stage results in a high-level representation of these requirements, such as the Entity-Relationship diagram. The focal point of the diagram is a set of entities that represent or model a particular information aggregate specified in the requirement [11]. Entity sets are a group of entities of the same type. Entities are described by attributes that provide

detailed information about the entity. One or more of the attributes will serve as an identifier (key) to distinguish different instances of the entity. Relationships between the entities depict the functional aspects of the information represented by the entities. The steps involved in arriving at the Entity-Relationship diagram are:

1. Selection of entities
2. Selection of relationships between entities
3. Selection of entity attributes
4. Identification of key attributes for entities

The above four steps were used during the conceptual stage to arrive at the conceptual/logical design of the GSRDB. The user's requirements were analyzed and consolidated to give one single global view. This view is called the enterprise view of the data, the "enterprise scheme". The enterprise scheme should be a representation of the real world and should be independent of storage and efficiency considerations [5]. The first step in arriving at the logical data base design is to select the entities which are of importance to the organization. This procedure is discussed in the next section.

2.6.1 Selection of Entities

An entity is a "thing" which can be distinctly identified [4]. Entities can be classified into different entity types, such as STUDENT and ADVISOR. In the E-R diagram, an entity type is represented by a rectangular-shaped box. According to the needs of the enterprise, entities can be classified into different entity types, such as STUDENT, ADVISOR, etc. Entities of the same type are classified as entity sets. For example, STUDENT is an entity so all STUDENT entities are members of the entity set STUDENT.

It is the responsibility of the data base designer to select the entity types that are important to the particular organization. This process was accomplished by analyzing the data elements, data values, and input and output documents that were identified in the requirements formulation and analysis step. The results of analyzing 12 input and 10 output documents and the 29 data elements are the 14 entities that are listed below. These entities are logical entities that are significant to the Computer Science Department and the design of this data base.

These entities are:

1. ADMITTANCE INFORMATION
2. ADVISOR
3. ADVISORY COMMITTEE
4. COURSE LIST
5. DEFICIENCY COURSE
6. MAJOR COURSE
7. MAJOR PROFESSOR
8. MASTERS RESEARCH
9. PROGRAM OF STUDY
10. STATUS
11. STUDENT
12. SUPPORTING COURSE
13. TEACHING ASSISTANT
14. UNIVERSITIES

A description of these entities follows:

1. ADMITTANCE INFORMATION

This entity was established based on the Department's desire to forecast, by semester and year, the number of applicants desiring to enroll in the Computer Science Department, and to be made aware of a student's goals and program, e.g., Ph.D., M.S. or special status.

2. ADVISOR

This entity was established because each student enrolled in the Computer Science Department is assigned an advisor. The Department is responsible for making this assignment and insuring that it is made in a

timely manner. The advisor counsels and advises the student about his program of study and monitors student progress for the Department.

3. ADVISORY COMMITTEE

M.S. and Ph.D. students require advisory committees. The student selects the committee, but the Department insures that a committee exists and records and approves of the membership.

4. COURSE LIST

This entity exists because the Department offers 68 courses. These courses may appear on three different input forms and participate in four separate relationships among entities that are of concern to the Department.

5. DEFICIENCY COURSE

This entity exists because the Department must keep track of the deficiency courses each student has to complete. Successful completion of these courses are as important in attaining the master's degree as the courses appearing on the program of study.

6. MAJOR COURSE

This entity exists because each student must complete a set of major courses to complete the master's degree.

7. MAJOR PROFESSOR

This entity stems from the Department's responsibility to insure that when students reach the point in their studies to consider their master's research, sufficient qualified professors are available to assist them in pursuing a research option. This entity will allow the

Department to monitor this assignment and the students to major professor ratio.

8. MASTERS RESEARCH

This entity exists because there is a considerable amount of information that is vitally important to both the student and the Department concerning master's research. This entity will enable the Department to be aware of the following:

- a. Research option (thesis, report, paper)
- b. The date the student submits his proposal
- c. Whether and when the proposal was approved
- d. The title of the research
- e. When a student's oral examination is scheduled
- f. The results of the oral examination (passed, failed conditionally passed)

9. PROGRAM OF STUDY

This entity exists because it is the major contract between the student and the Department. The Department agrees that a master's degree will be awarded to the student if he/she successfully completes the requirements listed on the program of study. This entity allows the Department to record this vital information.

10. STATUS

This entity exists because it is important to the Department to keep track of a student's current and past status. The student's status can change any number of times after his initial acceptance status. This entity enables the Department to keep track of the following:

- a. If a student changes from or to any of following:
 - 1) Regular
 - 2) Provisional
 - 3) Special
 - 4) Probationary
- b. The date of each change

11. STUDENT

The entity, student, exists to enable the Department to keep a record of descriptive information on each student. This entity is basic to the data base design and is directly related to nine of the twelve relations in the E-R diagram.

12. SUPPORTING COURSE

This entity exists because a student may have a number of courses that are completed outside the Department which are a part of the program of study.

13. TEACHING ASSISTANT

This entity enables the Department to maintain information concerning the computer science courses a student indicates he/she is qualified to teach.

14. UNIVERSITIES

This entity provides the Department with information concerning a student's previous education. The items of interest to the Department are as follows:

- a. Previously attended schools
- b. The dates of attendance
- c. Curriculum
- d. GPA
- e. The school's grading scale
- f. The degree awarded (if any)
- h. The date the degree was awarded (if a degree was awarded)

Each of the fourteen entities listed above contain a set of attributes which have unique values to specifically describe the entity. The attributes of the entities will be presented in Section 2.6.3. Additionally, these entities are related to one another and form

relationships. It is the responsibility of the data base designer to analyze the entities and form these relationships. This process is explained in the next section.

2.6.2 Selection and Identification of Relationships

A relationship is an association between occurrences of entities (e.g., 1:1, 1:N, M:N) [11]. Relationships between entities depict the functional aspects of the information represented by the entities [11]. Like entities the data base designer's responsibility is to identify the relationship sets of interest to the enterprise. Different types of relationships may exist between different types of entities. A relationship set is a set of relationships of the same type. In the Entity-Relationship diagram, a relationship is represented by a diamond-shaped box with lines connecting the related entity sets. A 1:1 relationship exists between the entity sets if no other indication appears on the lines connecting the entities. For example, MAJOR PROFESSOR_STUDENT, the absence of any characters on the lines connecting these two entities indicates that there is at most one major professor and one student for each instance of this relationship set (see Figure 2.4). A relationship also exists between STUDENT and PROGRAM_OF_STUDY. This is also a 1:1 relationship; each student has one program of study and one program of study is associated with one student. This relationship set is defined on two entity types (entity sets for multiple instances), STUDENT and PROGRAM OF STUDY (see Figure 2.9). A relationship set may also be defined on more than two entity sets, for example, PROGRAM OF STUDY, MAJOR COURSE, SUPPORTING COURSE, and DEFICIENCY COURSE. This relationship set is defined on four entity

types (sets)--PROGRAM_OF_STUDY, MAJOR_COURSE, SUPPORTING_COURSE, and DEFICIENCY_COURSE (see Figure 2.13). As stated earlier, a relationship is an association between occurrences of entities; therefore, 1:N, N:1, M:N relationships may exist. When these relationships exist, 1:N, N:1, and M:N appear on the lines connecting the entities. For example, 1:N appears on the lines connecting the STUDENT_UNIVERSITIES relationship (see Figure 2.6). This indicates that for each student there may be N instances of the entity university. An M:N relationship in the TEACHING_ASSISTANT_COURSES relationship is depicted (see Figure 2.11). This means that there is a many-to-many relationship between the entities COURSE LIST and TEACHING ASSISTANT. Any occurrence of the entity TEACHING ASSISTANT may have N occurrences of the entity COURSE LIST.

As stated earlier, an entity represents a real world concept about which information is recorded; a relationship is an explicit indication of how an entity is related to another. The relationship is as important and as definable as any entity or attribute of an entity. Connecting the entities by forming relationships and indicating the type 1:1, 1:N, M:N is the next step in arriving at the E-R diagram. This produces a view of how the entities are logically related. The relationships listed below were arrived at by first identifying and understanding each of the 14 entities and then logically relating them to one another. These relationships are described in Figures 2.2 through 2.13.

<u>Relationship</u>	<u>Figure</u>
1. STUDENT_ADMITTANCE INFORMATION	2.2
2. STUDENT_ADVISOR	2.3
3. STUDENT_MAJOR PROFESSOR	2.4
4. STUDENT_ADVISORY COMMITTEE	2.5
5. STUDENT_UNIVERSITIES	2.6
6. STUDENT_STATUS	2.7
7. STUDENT_MASTERS RESEARCH	2.8
8. STUDENT_PROGRAM OF STUDY	2.9
9. STUDENT_TEACHING ASSISTANT	2.10
10. TEACHING ASSISTANT_COURSE LIST	2.11
11. MAJOR, SUPPORTING, DEFICIENCY_COURSE LIST	2.12
12. PROGRAM OF STUDY_MAJOR, SUPPORTING, DEFICIENCY	2.13

As previously stated, the enterprise schema views the data from the whole enterprise point of view. This schema is a representation of the real world and is independent of storage, efficiency, and other implementation criteria. This view combines all the entity sets and relationship sets that are explained in Figures 2.2 through 2.13. The initial enterprise schema for this data base is shown in Figure 2.14.

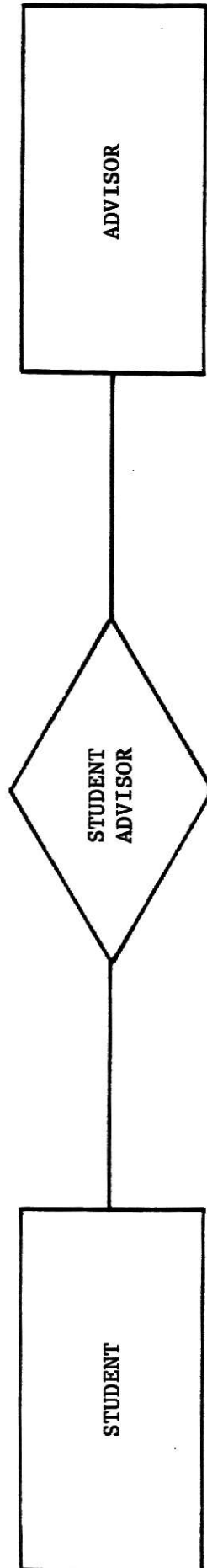
The next section explains the procedure for selecting attributes and evaluating the value types of the attributes. During this procedure, entities and relationships were reevaluated necessitating several changes. A discussion and the outcome of these changes appear in the next section.

NOTE: The next section discusses entity attributes, values, value sets and identifiers. It is suggested that the reader take a moment to examine Figure 2.15 to become familiar with the E-R diagrammatic technique before continuing to the next section.



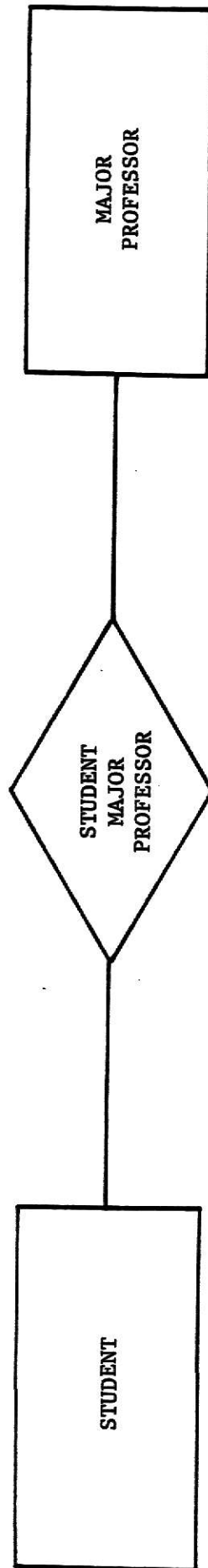
The STUDENT_ADMITTANCE INFORMATION relationship is formed by the entity sets STUDENT and ADMITTANCE INFORMATION. This relationship is a 1:1 mapping. That is, each STUDENT entity may be associated with only one instance of an ADMITTANCE INFORMATION entity.

Figure 2.2
STUDENT_ADMITTANCE INFORMATION Relationship



The STUDENT ADVISOR relationship is formed by the entity sets STUDENT and ADVISOR. This relationship is a 1:1 mapping. That is, each STUDENT entity may be associated with only one instance of the ADVISOR entity.

Figure 2.3
STUDENT_ADVISOR Relationship



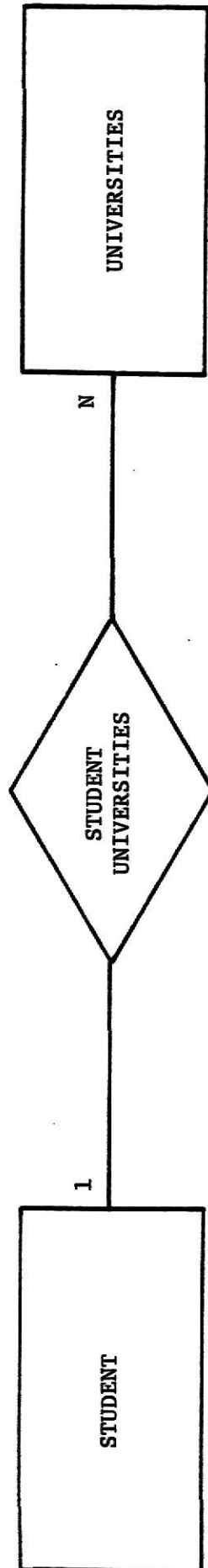
The STUDENT MAJOR PROFESSOR relationship is formed by the entity sets STUDENT and MAJOR PROFESSOR. This relationship is a 1:1 mapping. That is, each STUDENT entity may be associated with only one instance of the MAJOR PROFESSOR entity.

Figure 2.4
STUDENT MAJOR PROFESSOR Relationship



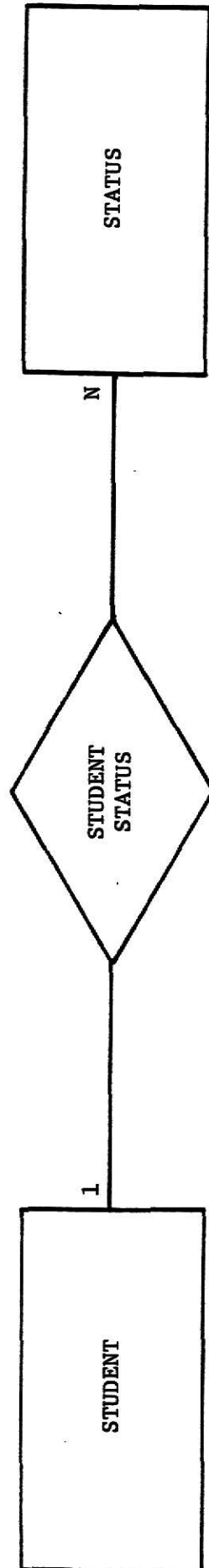
The STUDENT ADVISORY COMMITTEE relationship is formed by the entity sets STUDENT and ADVISORY COMMITTEE. This relationship is a 1:1 mapping. That is, each STUDENT entity can be associated with one instance of the entity ADVISORY COMMITTEE.

Figure 2.5
STUDENT ADVISORY COMMITTEE Relationship



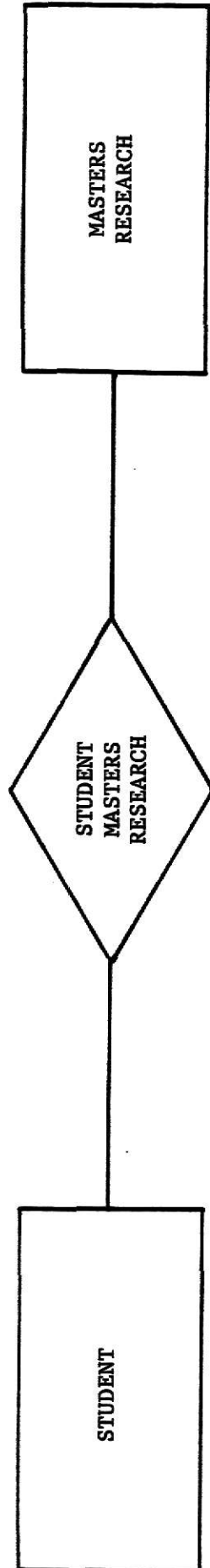
The **STUDENT_UNIVERSITIES** relationship is formed by the entity sets **STUDENT** and **UNIVERSITIES**. This relationship is a 1:N mapping. That is, each **STUDENT** entity can be associated with **N** instances of the **UNIVERSITIES** entity.

Figure 2.6
STUDENT_UNIVERSITIES Relationship



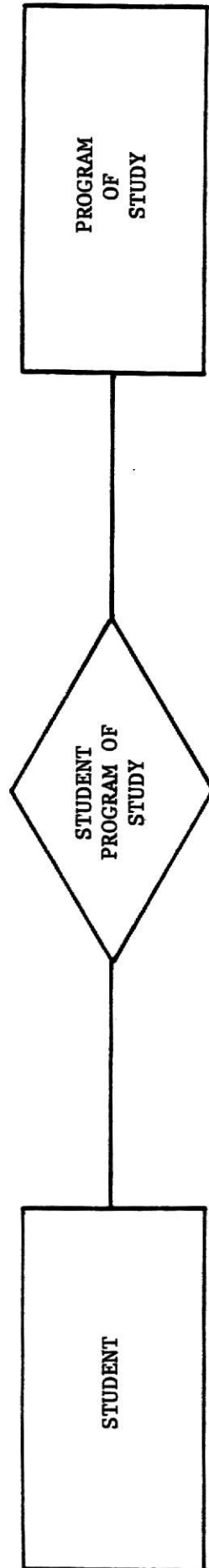
The **STUDENT_STATUS** relationship is formed by the entity sets **STUDENT** and **STATUS**. This relationship is a 1:N mapping. That is, each **STUDENT** entity can be associated with N instances of the **STATUS** entity.

Figure 2.7
STUDENT_STATUS Relationship



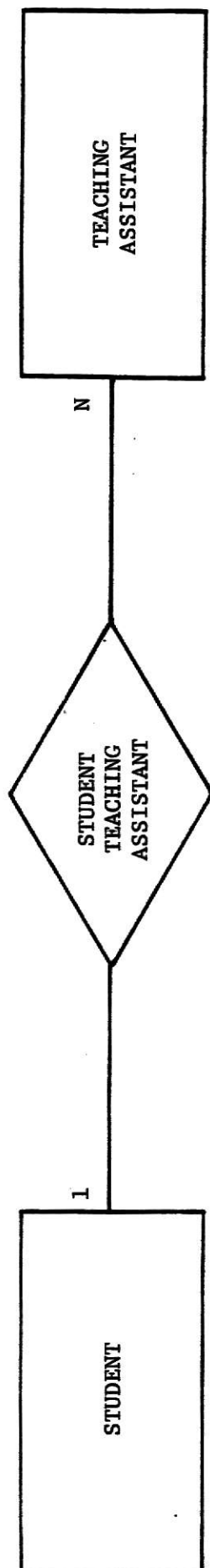
The STUDENT MASTERS RESEARCH relationship is formed by the entity sets STUDENT and MASTERS RESEARCH. This relationship is a 1:1 mapping. That is, each STUDENT entity is associated with only one instance of the MASTERS RESEARCH entity.

Figure 2.8
STUDENT_MASTERS RESEARCH Relationship



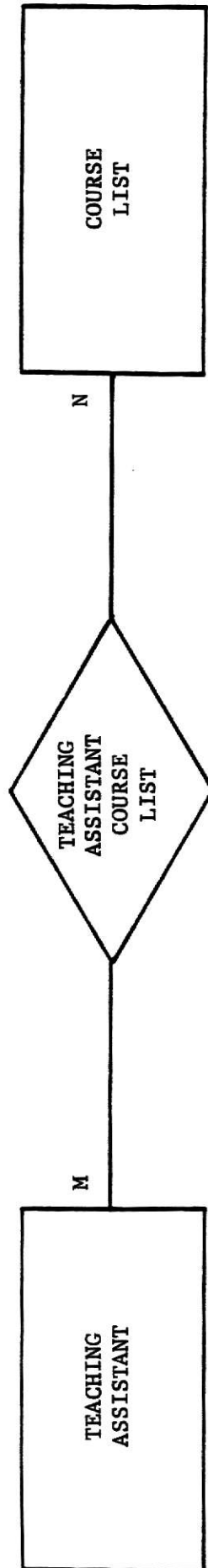
The STUDENT_PROGRAM OF STUDY relationship is formed by the entity sets STUDENT and PROGRAM OF STUDY. This relationship is a 1:1 mapping. That is, each STUDENT entity is associated with only one instance of the PROGRAM OF STUDY entity.

Figure 2.9
STUDENT_PROGRAM OF STUDY Relationship



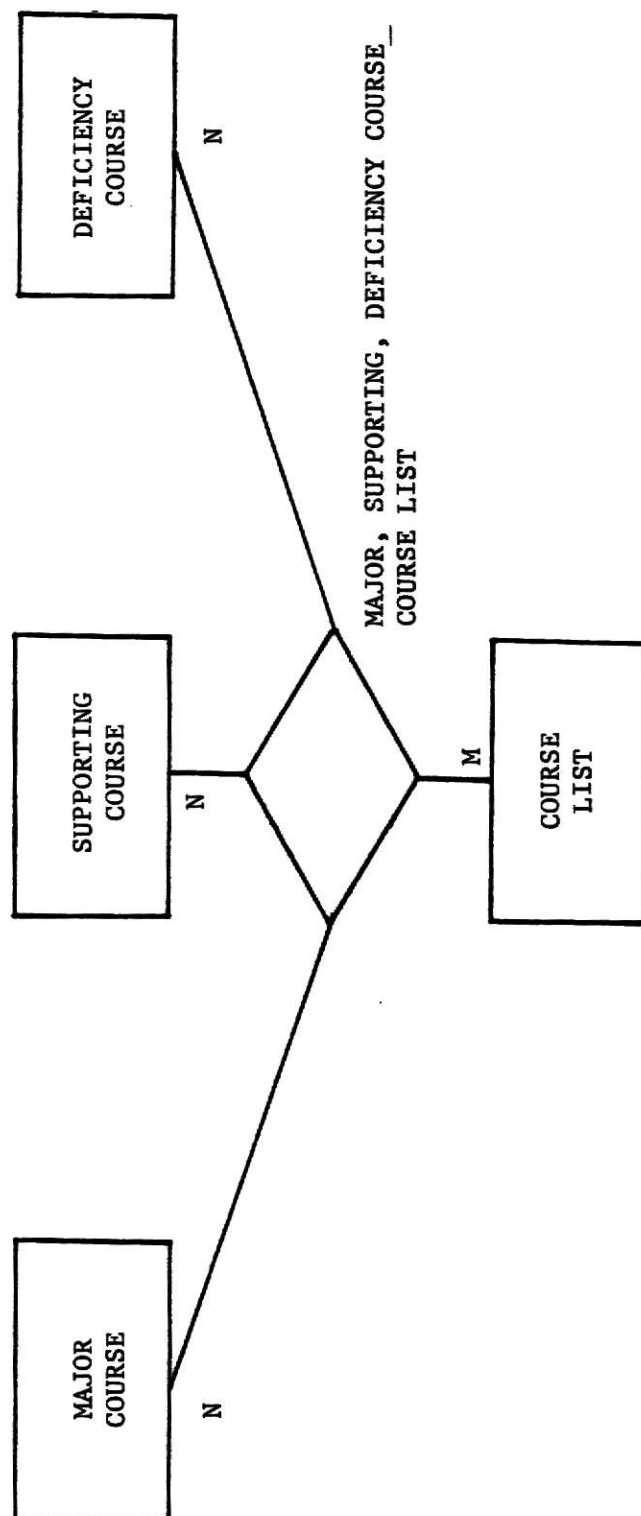
The STUDENT-TEACHING ASSISTANT relationship is formed by the entity sets STUDENT and TEACHING ASSISTANT. This relationship is a 1:N mapping. That is, each STUDENT entity can be associated with N instances of the TEACHING ASSISTANT entity.

Figure 2.10
STUDENT_TEACHING ASSISTANT Relationship



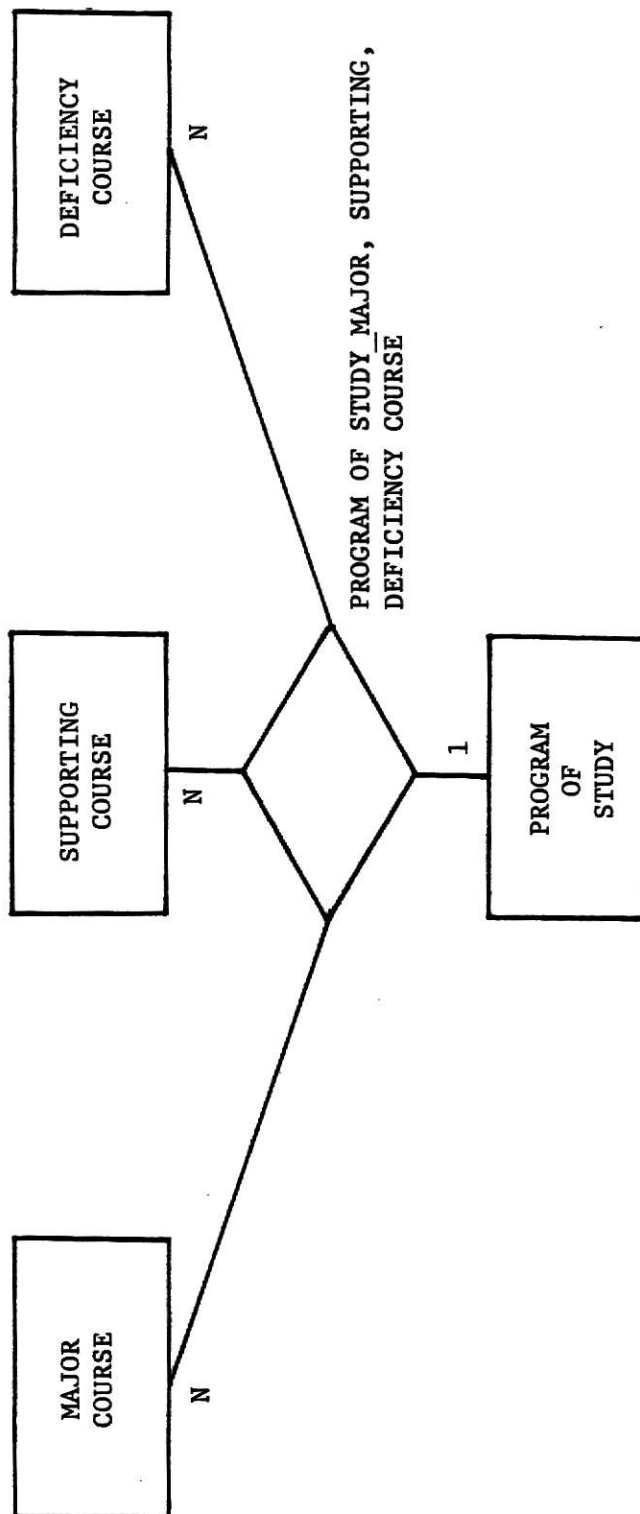
The TEACHING ASSISTANT COURSE LIST relationship is formed by the entity sets TEACHING ASSISTANT and COURSE LIST. This relationship is a M:N mapping. That is, each TEACHING ASSISTANT entity can be associated with one or several instances of the entity COURSE LIST. Additionally, any COURSE LIST entity could be associated with one or several instances of the TEACHING ASSISTANT entity.

Figure 2.11
TEACHING ASSISTANT COURSE LIST Relationship



The MAJOR, SUPPORTING, DEFICIENCY COURSE LIST relationship is formed by the entity sets MAJOR COURSE, SUPPORTING COURSE, DEFICIENCY COURSE AND COURSE LIST. This relationship is a M:N mapping. That is, any instance of MAJOR COURSE, SUPPORTING COURSE or DEFICIENCY COURSE can be associated with one or several instances of the COURSE LIST entity. Additionally, any COURSE LIST entity can be associated with one of several MAJOR, SUPPORTING, or DEFICIENCY COURSE entities.

Figure 2.12
MAJOR, SUPPORTING, DEFICIENCY COURSE LIST Relationship



The MAJOR, SUPPORTING, DEFICIENCY COURSE PROGRAM OF STUDY relationship is formed by the entity sets MAJOR COURSE, SUPPORTING COURSE, DEFICIENCY COURSE and PROGRAM OF STUDY. This relationship is a 1:N mapping. That is, each PROGRAM OF STUDY entity may be associated with one or several instances of the entities MAJOR COURSE, SUPPORTING COURSE, or DEFICIENCY COURSE.

Figure 2.13
PROGRAM OF STUDY MAJOR, SUPPORTING, DEFICIENCY COURSE Relationship

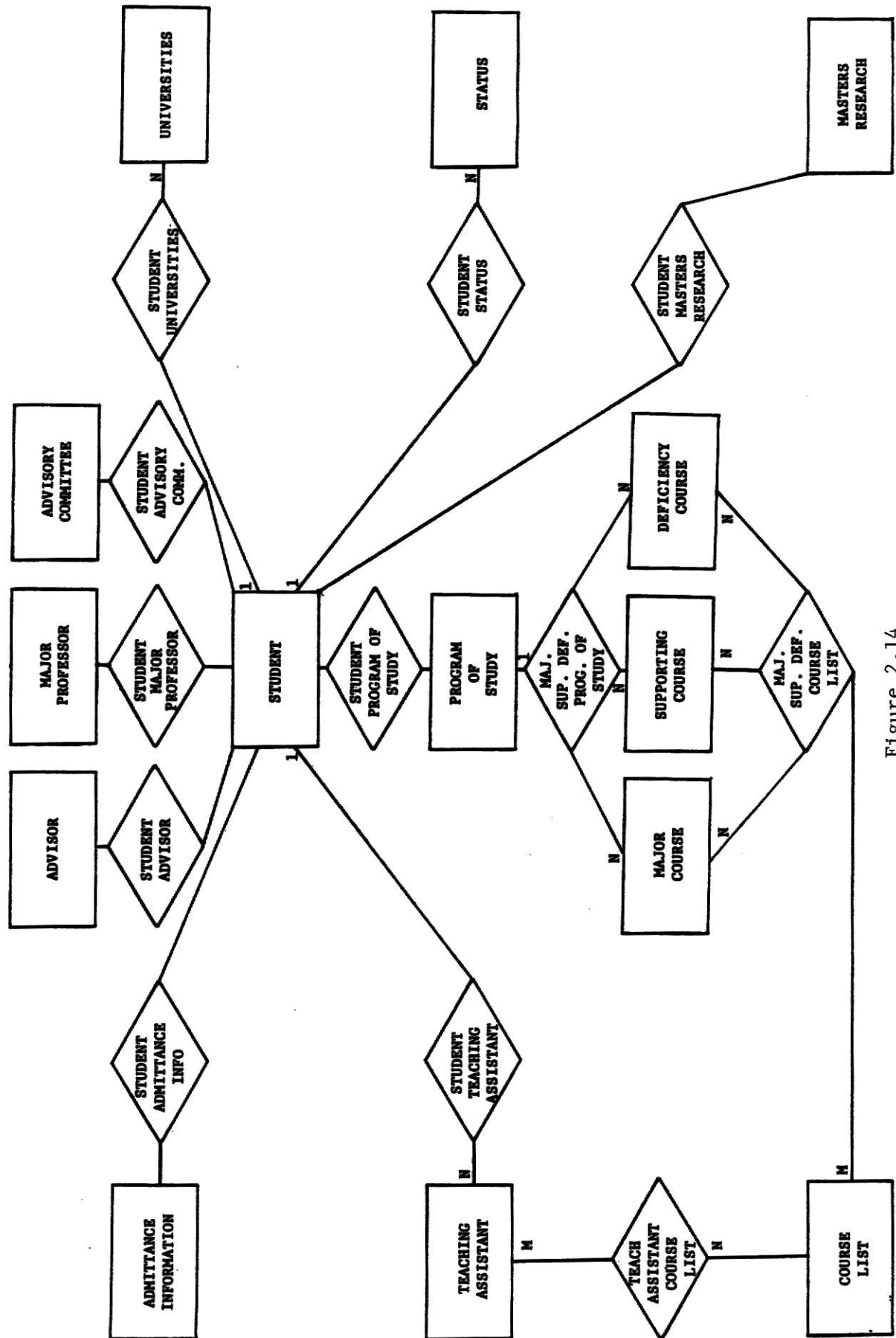
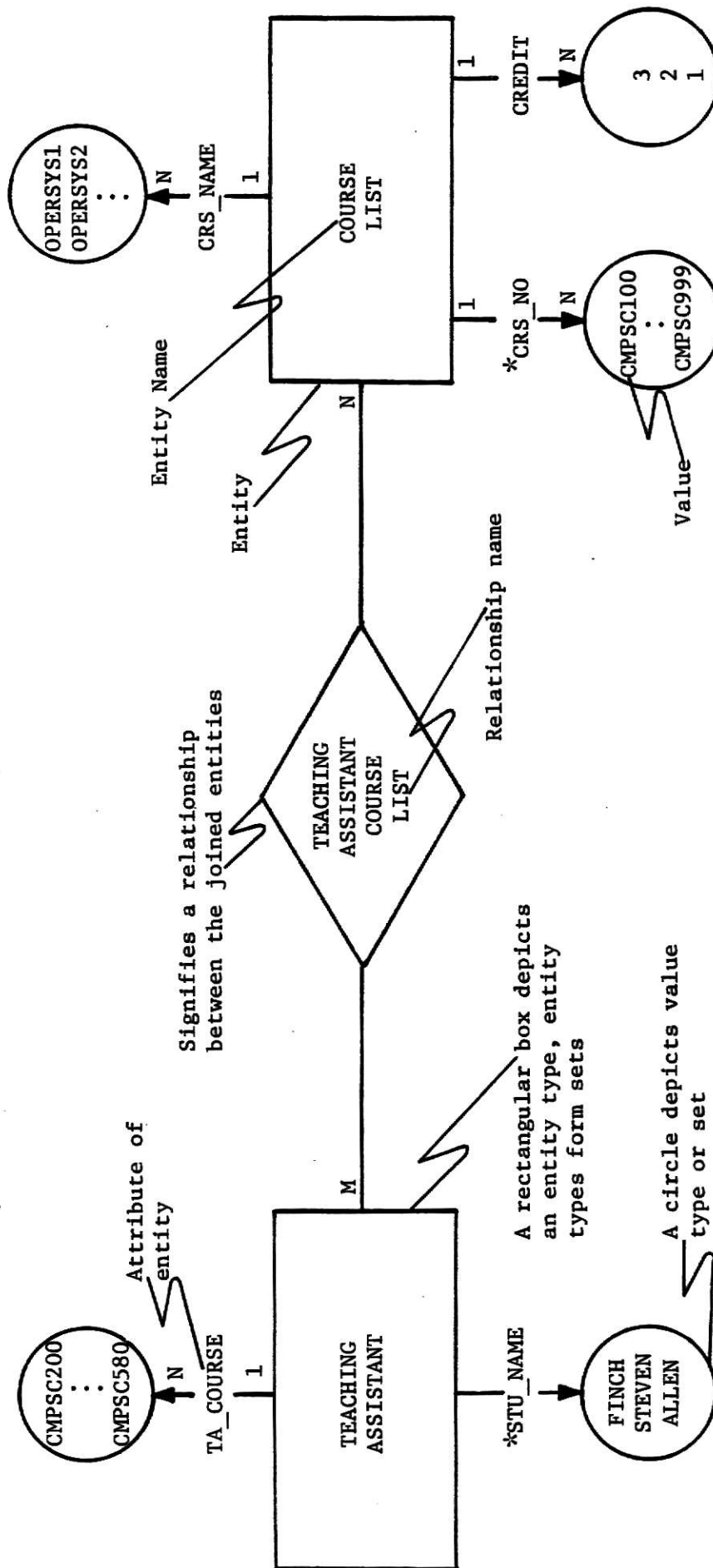


Figure 2.14
Initial GSRDB E-R Diagram



1:N next to the attribute arrow indicates a multi-valued attribute.
 1:N, M:N, 1:1 (no characters on the line) indicates the type of association of occurrences between the entities.
 * indicates key attribute (entity identifier(s)).

Figure 2.15
 E-R Diagrammatic Technique Example

2.6.3 Selection of Entity Attributes and Value Types

Entities and relationships have properties, which can be expressed in terms of attribute-value pairs [5]. For example, in the statement "the SSAN OF STUDENT X IS 149367712", "SSAN" is an "attribute" of student X and "149367712" is the value of the attribute SSAN. Values can be classified into different value types such as soc-sec-no., quantity, and color. In the E-R diagrammatic notation, a value type is represented by a circle (see Figure 2.15) and an attribute is represented by an arrow directed from an entity type to the desired value type.

In some cases, an attribute may have more than one value for a given entity. For instance, "CRS_NO" in the entity COURSE_LIST has several values, one value (CRS_NO) for each course taught by the Computer Science Department. In this case, the notation "1:N" is put next to the arrow to indicate that it is a multi-valued attribute. This is similar to the "repeating group" concept in a network or hierarchical data base model. However, many attributes, such as "SSAN" and "SEX", are single-valued. For simplicity, a "1:1" notation next to the attribute arrows is not used. Therefore, the absence of notation next to an attribute arrow indicates a 1:1 relation, one value for the value type.

So far I have considered only the attributes of entities. Sometimes we are also interested in the properties of a relationship. For instance, we may want to know the total number of SUPPORTING COURSE and MAJOR COURSE credit hours for a particular student. MAJOR COURSE, SUPPORTING COURSE credit hours is an attribute of the relationship MAJOR, SUPPORTING, DEFICIENCY_PROGRAM OF STUDY. The concept of

"attribute of relationship" is important in understanding the semantics data [5]. The concept is similar to the "relationship data" in "network" (CODASYL) type data base systems, and similar to the "intersection data" in hierarchical-type (IMS-TYPE) data base systems [5].

Previously I stated, attributes and their values are properties that describe the entities. The GSRDB design resulted in the selection of 90 attributes to identify the entities. These attributes and their values, as they relate to the entities, are found in Appendix A (the initial entity diagrams) and Appendix B (the final diagrams).

During the process of selecting these attributes, two entities were found to have only one attribute and another entity two attributes. Therefore, it was decided to eliminate these entities and make them attributes of two other entities. This decision reduced the data base size by reducing both the number of data structures and relations and eliminating the necessity for a key value for those relations. The eliminated entities are:

<u>Eliminated Entity</u>	<u>Attribute(s)</u>
ADVISOR	NAME (see Figure A.2)
MAJOR PROFESSOR	NAME (see Figure A.3)
ADVISORY COMMITTEE	MEMBERS (2) (see Figure A.4)

The advisor name was made an attribute of the STUDENT entity (see Figures B.1, Appendix B) and the name of the major professor and committee were made attributes of the MASTERS RESEARCH entity (see Figure B.7, Appendix B).

Additionally, the entities MAJOR COURSE, SUPPORTING COURSE, DEFICIENCY COURSE and PROGRAM OF STUDY were eliminated. These entities were eliminated based on the following:

1. MAJOR COURSE, SUPPORTING COURSE, DEFICIENCY COURSE (see Figure A.9, Appendix A):

Attributes for these three entities were identical. Therefore, I decided to merge these three entities and establish a new entity called GRAD_COURSES. Three attributes, MAJOR, SUPPORT, and DEF, were added to the entity to indicate whether the course was a major, supporting or deficiency course. This decision reduced the data base size by reducing both the number of data structures and relations and eliminated the necessity for a key value for each of the relations. Additionally, this decision supported a user friendly transaction processing system by simplifying the input requirements. The type course is indicated by merely placing an X next to the appropriate value (see Figure B.2, Appendix B).

2. PROGRAM OF STUDY:

The attributes for this entity were arrived at by analyzing the program of study form. However, after careful consideration as to where these attributes should logically appear, the entity was eliminated and the attributes reassigned to STUDENT_INFO and MASTERS_RESEARCH. This decision was based on the logical association of these attributes with the attributes of these two entities. This decision reduced the data base size by reducing both the

number of relations and data structures and the necessity for a key value for this relation (see Figure A.12, Appendix A).

The elimination and the merging of the entities above resulted in designing a new enterprise schema for the Department. The final enterprise schema is shown in Figure E.1, Appendix E.

Normally there will be several instances of each entity; therefore, some scheme must exist to uniquely identify each instance of an entity. The method of doing this is discussed in the next section.

2.6.4 Selection of Key Attributes for Entities

As stated earlier, an entity is a real world concept that is of importance to the organization for which the data base is being designed. Information is gathered and recorded for each instance of the entity. It is vitally important to be able to uniquely identify each instance of an entity. In most cases, entities will become relations in the relational data model; therefore, like a key in a relation, each entity must have an "ENTITY IDENTIFIER" selected to positively identify each instance of the entity. This is accomplished by selecting an attribute or a combination of attributes that will contain unique values for each instance of the entity. For example, the entity COURSE_LIST has three attributes CRS_NO, CRS_NAME, and CREDIT. Since all Computer Science Department courses have a unique course number the "ENTITY IDENTIFIER" for this entity is CRS_NO (see Figure B.3, Appendix B).

CHAPTER III

DATA BASE MODELS

3.0 Data Base Systems and Models

There are many data base systems in use at the moment. They can be classified in three (3) major categories: hierarchical, network and relational. One of the major differences among them is the type of logical data structures which can be supported. Hierarchical data base systems, such as IBM's Information Management System (IMS), require data record types to be organized in a hierarchical form. This hierarchical data structure works well with some data bases, but it becomes difficult to design data bases using a hierarchical data base system when a natural hierarchy among record types does not exist. Network (or CODASYL) data base systems, such as Honeywell's Integrated Data Store (IDS), UNIVAC's DMS-1100, and Cullinane's IDMS, provide more complex data structure capabilities than the hierarchical data base systems [5]. For example, network data base systems allow a record type to have multiple record types as its "parents" [5]. Relational systems use tables as the logical data structures. The relational model maintains logical simplicity of the data structure through the use of flat files. A detailed discussion of the relational model is presented in the next section.

3.1 The Relational Model

The relational data base approach was conceived by E. F. Codd in 1970. It is a significantly different approach to the logical

description and manipulation of data than the previously mentioned models. It strives to avoid many of the drawbacks found in both the network and hierarchical data base systems. It views the logical data base as a simple collection of two dimensional tables called relations [2]. A data structure for these relations are flat files. A flat file is one in which each record instance has a similar number of fields. The conventional COBOL-like logical file structure without repeating groups is a flat file [2]. These tables are easily understood and handled by users with little or no training in programming, and involve no consideration of positional, pointer or access path aspects [2].

A relation or table is a two-dimensional array with the following characteristics:

1. Each entry in the table is a single attribute; there are no repeating groups. Each attribute must be a simple attribute; that is, it must not represent another relation. A relation is said to be normalized (flat) if it is composed of simple attributes; otherwise, it is said to be unnormalized.
2. Each column of the table is assigned a distinct name, an attribute name, and is made up of values for the same data item.
3. All rows or tuples are distinct; duplicates are not allowed [2].
4. The rows and the columns can be ordered in any sequence at any time without affecting the information content or the semantics [2].

A relational data base is viewed by programmers and users as a time-varying collection of normalized relations of various degrees, manipulated by powerful operators for extracting rows or columns and joining them into new relations.

Each tuple of a relation must have a key by which the tuple can be uniquely identified and differentiated from the other tuples of that

relation. The key is either a simple attribute or combinations of attributes. A key made up of a combination of attributes is nonredundant if no attribute of the key can be deleted without destroying the ability to uniquely identify each tuple. There may be more than one set of domains which may be a key; that is, that uniquely identifies each tuple and which is nonredundant. These sets are called candidate keys [2]. The primary key is the set of domains chosen to identify the tuples. Null values are possible for certain instances of the relation; however, no component of a primary key may be null. This rationale stems from the fact that all entities must be distinguishable from one another. Since primary keys perform this unique identification of instances of the relation, a primary key that was wholly or partially null would be a contradiction of this basic rule.

The relational approach to data base management system design is attractive for a number of reasons. These attractive features are presented in the next section.

3.1.1 Advantages of the Relational Approach

The relational approach to data base management systems has several significant advantages over both the network and hierarchic approach (especially when the relations are of third normal form, addressed in Section 3.2). These advantages are:

1. Simplicity and ease of use. Relations (two-dimensional tables) are undoubtedly easier to understand and deal with for the large number nonprogramming oriented users. Users are presented with a single and consistent data structure; therefore, there is no undue concern for 1 to N versus M to

N relationships. Users formulate their requests via powerful languages (e.g., QUEL) in terms of information content with no reference to storage oriented access paths. The response to queries is a table format with each tuple providing one unique instance of information (record) [2].

2. Data independence. The immunity of application programs to changes in the data model, submodel, and physical organization and physical access paths is greatly enhanced. The user does not have to be concerned with physical structure and access paths, such as physical position and pointers. Additionally, a higher degree of data independence can be achieved easier with relations than with most tree or network structures [2].
3. Symmetry. In relational models all information is represented by data values and there is no preferred type of question at the user interface level. Whereas data models involving connections between records may give undue preferential treatment to those access requests matching the connected structures of the data base. Relational data bases can be designed to accommodate frequent use of a certain type query and query content through a certain type of indexing inversion [2].
4. Flexibility. Powerful operations such as join and projection are derivations of (from a user's point of view, not machine performance criteria) a wide variety of new

tables. This flexibility permits satisfaction of a wide variety of users and their changing needs.

5. Good theoretical foundation. The relational model relies on the well-developed mathematical theory of relations and on the so-called first order predicate calculus. This theoretical base permits the definition of relational completeness and the formal study of good data base design (in terms of normalization). The mathematical background will not necessarily burden less sophisticated users and the benefits of a sound foundation are inherent [2].

These characteristics of the relational approach are very significant, so much so that most of the data base research since 1970 into such areas as concurrency, locking, security, integrity, and view definition have taken the relational approach as a starting point [6]. However, there are a few challenges that the relational approach must face. These challenges are presented in the following section.

3.1.2 Challenges of the Relational Approach

The relational data base approach is attractive and promising. However, there are no commercial fully relational generalized data base management systems at the present time. The relational approach must achieve a cost-effective commercial implementation. At present the major problem is performance. Most of the computer architecture that is currently being used is not suitable to support a relational data base management system. Relational operators, such as the join operation, may result in large tables whose extension requires large amounts of

storage. One of the most challenging tasks of the relational approach is the practical and efficient implementation of this operation [2]. Reductions in undesired side effects, anomalies, are achieved by constructing relations in 3NF. This normalization process requires a considerable amount of duplication of keys. Therefore, there may be a considerable proliferation of storage redundancy of key attributes as we step through the normalization procedure. While this logical redundancy does not necessarily have to be real, a significant amount of overhead in the system would be required to eliminate it.

Continued interest and research in the relational approach has precipitated research and advances in hardware technology to support relational data base management systems. As this hardware becomes commercially available, a cost-effective commercial implementation of a relational data base management system should be achieved.

Normalization is of great concern in relational data bases. This process is presented in the next section.

3.2 Normalization and Functional Dependency

Normalization is the process by which any logical data base structure, such as a network or hierarchic data base, can be transformed by a data base designer into a set of normalized relations, that is, a set of flat relations that have no repeating groups. Just as a network data base can be transformed into a hierarchic data base by introducing redundancy, so can any data base be transformed into a relational data base by introducing additional redundancy [2].

The classification of a relation into first, second, or third normal form rests on the functional dependencies existing among the attributes or domains particular to a relation [2]. Functional dependencies are directly determined by the meaning or semantics of the data contents as interpreted by a data base designer [2].

Given a relation R , the attribute B is said to be functionally dependent on attribute A if at every instant of time each value of A has no more than one value of B associated with it in the relation R [2]. Stating that B is functionally dependent on A is equivalent to stating that A identifies or determines B which may be denoted as $A \rightarrow B$ [2]. This is in accord with mathematical logic in which $A \rightarrow B$ means A identifies B , that is, if A has a certain value " a " then B must have a value " b " [2].

Functional dependencies between attributes are established strictly by the meaning of the data. For example, in the relation `COURSE_LIST` (`CRS_NO`, `CRS_NAME`, `CREDIT`), saying that `CRS_NAME` is functionally dependent on `CRS_NO` means that each given course identified by `CRS_NO` must have only one name. Clearly, in every relation every nonkey attribute is functionally dependent on at least the key attribute [2]. If a relation has more than one key attribute, then all its attributes are dependent on each key attribute. An attribute can be functionally dependent on a group or collection of attributes.

Therefore, an attribute or collection of attributes, B , of a relation R is said to be fully functionally dependent on another collection of attributes, A , of relation R if B is functionally dependent on the whole of A but not on any subset of A [2]. This is an important concept because in order for a relation to be in second normal

form (2NF), all nonkey domains must be fully functionally dependent on the primary key. For example, in the relation GRAD_COURSES (NAME, CRS_NO, SEM_YR, CRS_GRD, DEF, REPEAT, MAJOR, SUPPORT), B = (SEM_YR, CRS_GRD, DEF, REPEAT, MAJOR, SUPPORT), R = the relation GRAD_COURSES and A = NAME, CRS_NO (the concatenated key). Each attribute contained in B, SEM_YR, CRS_GRD, etc., is fully functionally dependent on both NAME and CRS_NO. Therefore, this relation meets the requirements for 2NF.

The motivation for normalization and the interest in using third normal form (3NF) relations arises as a result of observing that in a dynamic data base environment possibly undesired side effects referred to as anomalies in relational jargon may result due to insertions, deletions, and updates [2]. A simple way to avoid these difficulties is to insure that only one "concept" or "entity" is represented in each relation. Anomalies can occur in relations that are in 2NF; therefore, every effort must be made to design the relations so that they are in 3NF. A relation R is said to be in 3NF if and only if the nonkey attributes (attributes not participating in the primary key) of R, if any, are both (1) mutually independent and (2) fully functionally dependent on the primary key of R [2].

Conversion to higher normal forms reduces the various anomalies cited. However, it can introduce two concerns: (1) duplicate recording of a key field of one of the split relations; and (2) as a result of duplicate recording, we now face the burden of policing interrelation consistency [2]. These potential problems are a result of splitting relations for normalization purposes. Splitting relations puts determinant attribute(s) stored in each relation with one of the attributes acting as the key in one of the relations. This potential

problem can be made less likely. As stated earlier, ADBG system was incorporated in the design process of this data base. This system implements Bernstein's algorithm [1] which produces a 3NF schema with the minimum number of relations for a given set of functional dependencies. Therefore, with fewer relations these problems will be less likely to occur.

Although we must contend with the potential problems of redundancy giving us storage concerns and interrelational consistency concerns, the concept of normalization and the desirability of achieving 3NF relations in a data base are of significant importance in data base design and use. The 3NF does not guarantee that anomalies will be prevented. However, they are much less likely to occur if the relations are formed based on one "concept" or "entity" which coincide with the 3NF definition.

The GSRDB design objective was to achieve 3NF. Therefore, the important issues of functional dependencies and normalization were given careful study. Understanding these concepts and the entities, attributes and their semantics provided the background information necessary to design the relations.

3.3 Automatic Data Base Generator System

Essentially, this system aids the system's analyst, or Data Base Administrator (DBA) in constructing a data base for a specific enterprise environment [9]. The system aids the analyst in obtaining a complete set of data elements and then permits certain manipulations and

tests on these elements. This system also provides a mechanism for easily incorporating changes during the design period.

Basically, this system automates the analysis of an organization's documents. The system views the organization's documents within the classification schema: Input, Output, or Resident. The organization's documents are entered into the system where various manipulations can take place to insure that all the data elements necessary to support an organization's operations are included in the data base.

After all documents have been entered into the system, the KEY command is used to declare the key column(s) in one or more documents. This will generate functional dependencies among the columns of the document. The PREPARE command is used to prepare functional dependencies in a form suitable for input to the Bernstein's algorithm (see Figure 3.1) currently implemented by the ADBG system at Kansas State University [9]. The execution is done in two steps. In the first step, abbreviated names, are assigned to all the column names that exist in the system. A dictionary of the column names with the associated abbreviated names is printed at the end of this step. The second step scans all the documents in the system and outputs a set of functional dependencies to a data file. These functional dependencies are formed by treating each document as a relation. The left-hand side is derived from all the key columns specified for any document. The other nonkey columns are used in the right-hand side of the functional dependency [9]. The system uses Bernstein's algorithm #2 to organize the attributes into relations which form a 3NF schema.

1. (Eliminate extraneous attributes.) Let F be the given set of FDs. Eliminate extraneous attributes from the left side of each FD in F , producing the set G . An attribute is extraneous if its elimination does not alter the closure of the set of FDs.
2. (Find covering.) Find a nonredundant covering H of G .
3. (Partition.) Partition H into groups such that all of the FDs in each group have identical left sides.
4. (Merge equivalent keys.) Let $J = \emptyset$. For each pair of groups, say H_1 and H_2 , with left sides X and Y , respectively, merge H_1 and H_2 together if there is a bijection $X \leftrightarrow Y$ in H^+ . For each such bijection, add $X \rightarrow Y$ and $Y \rightarrow X$ to J . For each $A \in Y$, if $X \rightarrow A$ is in H , then delete it from H . Do the same for each $Y \rightarrow B$ in H with $B \in X$.
5. (Eliminate transitive dependencies.) Find an $H' \subseteq H$ such that $(H' + J)^+ = (H + J)^+$ and no proper subset of H' has this property. Add each FD of J into its corresponding group of H' .
6. (Construct relations.) For each group, construct a relation consisting of all the attributes appearing in that group. Each set of attributes that appears on the left side of any FD in the group is a key of the relation. (Step 1 guarantees that no such set contains any extra attributes.) All keys found by this algorithm will be called synthesized. The set of constructed relations constitutes a schema for the given set of FDs.

Steps 1 through 4 are effectively implemented as in Algorithm 1. Step 5 can be effectively implemented by using the membership algorithm. Algorithm 2 can then be implemented in the same $O(L^2)$ time bound as Algorithm 1.

Figure 3.1
Bernstein's Algorithm #2

3.4 Relations

The ADBG system became available for use on this project after the requirements formulation and analysis step and the initial design of the relations were complete. The ADBG system was used to check functional dependencies and normalization of the relations and assist in preparing the data dictionary for this data base.

3.4.1 Initial Design

The initial design of relations was made based on the entities, their attributes, the functional dependencies of the attributes and in support of the transaction processing system. Support of the transaction processing system for the design of the relations required that the relations be constructed to support each screen display. This entailed insuring that the relations not only conformed to 3NF criteria but also to the constraints imposed by the screen data requirements.

The initial design of the relations was then provided as input to the ADBG system. The results of this procedure are presented in the next section.

3.4.2 ADBG Design I

The following nine relations were provided as input to the ADBG system (see Appendix C):

1. COURSE_LIST
2. GPA
3. GRAD_COURSES
4. INITIAL_ADM_INFO
5. MASTERS_RESEARCH
6. STUDENT_STATUS
7. STUDENT_INFO
8. TEACHING_ASSISTANT
9. UNIV_ATTENDED

These relations were processed in accordance with Bernstein's algorithm #2. The output of this program produces a data dictionary containing the names of 87 attributes and a 3NF schema of five relations. The following five relations were produced (see Appendix C):

1. COURSE_LIST
2. GRAD_COURSES
3. STUDENT_INFO
4. STUDENT_STATUS
5. UNIV_ATTENDED

The ADBG system changed the relations as follows:

Five relations (GPA, INITIAL_ADM_INFO, MASTERS_RESEARCH, STUDENT_INFO, and TEACHING_ASSISTANT) were merged into one relation and the attributes SPEC and CURRICULUM were removed from the relations STUDENT_STATUS and UNIV_ATTENDED, respectively (see Appendix C).

The five relations were merged together because each had the same left-hand side (LHS key), meaning that the attributes in each of the merged relations were all fully functionally dependent (FD) on the same key attribute, NAME. Additionally, each of the attributes in the merged relation were mutually independent and no transitive dependencies existed in any of the relations.

The attributes SPEC and CURRICULUM were removed from the relations STUDENT_STATUS and UNIVERSITIES_ATTENDED because the functional dependencies of these two attributes were defined as indicated below:

<u>Attribute</u>	<u>Relation</u>	<u>Functional Dependency</u>
SPEC	INITIAL_ADMIN_INFO	NAME --> SPEC
SPEC	STUDENT_STATUS	NAME, STATUS_DATE --> SPEC
CURRICULUM	STUDENT_INFO	NAME --> CURRICULUM
CURRICULUM	UNIV_ATTENDED	NAME, DATES_ATTENDED --> CURRICULUM

Relations STUDENT_STATUS and UNIV_ATTENDED both have extraneous attributes defining the functional dependency of attributes SPEC and CURRICULUM. Because STATUS_DATE and DATES_ATTENDED are extraneous attributes in defining SPEC and CURRICULUM, the relations STUDENT_STATUS and UNIV_ATTENDED were not in 2NF or 3NF. Therefore, the ADBG system removed these attributes (SPEC, CURRICULUM) from the relations.

The design of STUDENT_STATUS and UNIV_ATTENDED violated the uniqueness assumption in that for any two sets of attributes X and Y, there is at most one FD $X \rightarrow Y$. However, the semantics of the attributes SPEC and CURRICULUM were different in each of the relations; therefore, this problem was corrected by renaming the attributes as follows:

<u>Relation</u>	<u>Attribute</u>	<u>New Attribute</u>
STUDENT_STATUS	SPEC	SPEC
INITIAL_ADM_INFO	SPEC	I_SPEC
STUDENT_INFO	CURRICULUM	CURRICULUM
UNIV_ATTENDED	CURRICULUM	SUBJECT_AREA

Additionally, a survey of the UNIV_ATTENDED relation revealed that the attribute SCHOOL did not appear in the 3NF schema produced by the ADBG system. Cross-referencing this with the input relations and the data dictionary revealed that this attribute had not been included as input to the program. Modification of the relations is discussed in the next section.

3.4.3 Second Design

There were no changes to the LHS (KEY) of the relations. Therefore, it was anticipated that the relations (GPA, INITIAL_ADM_INFO,

MASTERS_RESEARCH, STUDENT_INFO, and TEACHING_ASSISTANT) would again be merged. The attributes were renamed as discussed in Section 3.4.2, and the attribute SCHOOL was added to the UNIV_ATTENDED relation. The results of these changes are explained in the next section.

3.4.4 ADBG Design II

As anticipated, the same five relations were merged together and the 3NF schema that was produced was identical to the first schema produced except that the new attributes added were contained in the appropriate relations. A review of the data dictionary indicated that the three new attributes I_SPEC, SCHOOL, and SUBJECT_AREA were added, resulting in a total of 90 attributes for the GSRDB (see Appendix D).

3.4.5 ADBG Commands

ADBG provides a number of commands that can assist the DBA or system's analyst in the design of a system. The additional commands used to aid in reviewing the design were:

1. PRINT DOCUMENT: This command causes all documents (relations) that exist in the system to be printed, ordered by the first letter of each relation name.
2. PRINT LIST: This command causes all relations with their attributes to be printed, ordered by the first letter of the relation. Additionally, the key attributes are indicated by an asterisk.
3. PRINT COLUMN: This command causes all columns (attributes) to be printed. The attributes are numbered and ordered by the first letter of the attribute name.
4. PRINT XREF: This command causes a cross reference of the columns (attributes) to the documents (relations) in the system to be printed. These are also ordered by the first letter of the attribute.

The data dictionary, 3NF schema, and listings provided by the commands described above are contained in Appendices C and D.

CHAPTER IV

GRADUATE RECORD DATA BASE DESIGN

4.0 Schema

The schema is the logical description of the data base. It includes the definition of the name and data type of each field making up each relation in the data base and also defines the relationships between the relations.

A relation in a relational data base actually has two components--an extension and an intension. The extension of a given relation is the set of tuples appearing in that relation at any given instant [6]. The extension varies with time, that is, it changes as tuples are created, destroyed, and updated. The intension of a given relation, by contrast, is independent of time. Basically, it is the permanent part of the relation; in other words, it corresponds to what is specified in the relational schema. The intension defines all permissible extensions [6]. More precisely, the intension is the combination of two things: a naming structure and a set of integrity constraints. The naming structure consists of the relation name plus the names of the attributes (each with its associated domain name). The schema designed for the GSRDB will be presented using the intension form for each relation. The GSRDB schema is:

COURSE_LIST (CRS_NO, CRS_NAME, CREDIT)

GPA (NAME, CURRENT_GPA)

GRAD_COURSES (NAME, CRS_NO, SEM_YR, CRS_GRD, DEF, REPEAT,
MAJOR, SUPPORT)

INITIAL_ADM_INFO (NAME, FALL, SPRING, SUMMER, YEAR, MS,
PHD, I_SPEC)

MASTERS_RESEARCH (NAME, THESIS, REPORT, PAPER, PROPOSAL_DATE,
PROPOSAL_APPROVED, TITLE, ORALS_DATE, PASS,
FAIL, COND_PASS, MAJ_PROF, MEMBER_2,
MEMBER_3)

STUDENT_STATUS (NAME, STATUS_DATE, REG, PROV, SPEC, PROB)

STUDENT_INFO (NAME, SSAN, LOCAL_ADDRESS, PERMANENT_ADDRESS,
LOCAL_TEL_NO, SEX, RESIDENCY_DATE, CITIZENSHIP,
PREVIOUS_NAME, GRAD_APP_DATE, GRE_OFFICIAL,
GRE_VERBAL_SC, GRE_QUANT_SC, GRE_ANALYTICAL_SC,
ADV_GRE_OFFICIAL, ADV_GRE_FLD, ADV_GRE_SC,
TOEFL_OFFICIAL, TOEFL_SC, TOEFL_DATE,
FINANCE_REQ, MEDICAL_REQ, LETTERS_OF_REC,
CURRICULUM, ADVISOR, PROG_ENTRY_DATE,
EXP_GRAD_DATE, PROG_OFF_STUDY_APP_DATE,
CS_INFO_DOC_DATE, CMPSC_APP_DOC_DATE,
PROC_FEES_PD)

TEACHING_ASSISTANT (NAME, CMPSC_200, CMPSC_201, CMPSC_202,
CMPSC_203, CMPSC_204, CMPSC_205, CMPSC_206,
CMPSC_300, CMPSC_305, CMPSC_306, CMPSC_405,
CMPSC_410, CMPSC_420, CMPSC_430, CMPSC_450,
CMPSC_560, CMPSC_580)

UNIV_ATTENDED (NAME, DATES_ATTENDED, SCHOOL, SUBJECT_AREA, GPA,
SCALE, DEGREE, DEGREE_DATE)

As stated earlier, the intension of a relation is the combination of both a naming structure and a set of integrity constraints. The naming structure, relation name, and attributes were presented above. The integrity constraints refer to the mechanisms used to preserve the validity, consistency, and accuracy of the preservation of the semantic definition of the data base, e.g., as expressed by the functional dependencies (FDs). Bernstein's algorithm #2 guarantees this format integrity. One of the objectives of data base technology is also to provide a means to maintain control over and preserve the integrity of the values within the data base. One basic way to accomplish this is to specify data types, formats and value ranges that are authorized for each attribute in the data base. These specifications would be contained in a data dictionary. The GSRDB data dictionary is discussed in the next section.

4.1 Data Dictionary

One of the most important DBA tools is the data dictionary. The data dictionary is effectively a data base in its own right--a data base that contains "data about data" (that is, descriptions of other objects in the system, rather than simply "raw data") [6]. Specifying the attribute name, field length, domain, value, and any constraints that may exist between values in relations will help the DBA maintain integrity in the data base.

Integrity mechanisms fall in two categories: data validity and consistency [2]. Validity refers to data types and format, value ranges, value ranges of related fields and uniqueness of a key field.

Consistency normally entails some constraint in some relationship between data objects in the data base [2]. Maintenance of data validity and consistency constraints requires a significant amount of overhead. However, it is necessary that each transaction see a consistent, correct view of the data base.

The GSRDB data dictionary contains the schema, attribute name, field length, domain, value range, and specifies constraints between the values of attributes in certain relations. The data dictionary is shown in Figure 4.1.

4.2 Design Considerations

The logical design stage has produced a relational schema that is independent of a specific DBMS. Although the schema is independent of a particular DBMS, it has been designed based on the following considerations:

1. Computer Science Department data requirements
2. Computer Science Department operational functions
3. Computer Science Department control and planning tasks
4. Computer Science Department operating policies
5. Support of a user friendly transaction processing system
6. Minimize the front-end programming requirements (the number of data manipulation commands required to move the data in and out of the data base)
7. User accessing requirements (ad-hoc queries)
8. Logical association of attributes (E-R diagrams)
9. Integrity and consistency

Schema

- I. COURSE_LIST (CRS_NO, CRS_NAME, CREDIT)
 - II. GPA (NAME, CURRENT_GPA)
 - III. GRAD_COURSES (NAME, CRS_NO, SEM_YR, CRS_GRP, DEF, REPEAT, MAJOR, SUPPORT)
 - IV. INITIAL_ADM_INFO (NAME, FALL, SPRING, SUMMER, YEAR, MS, PHD, I_SPEC)
 - V. MASTERS_RESEARCH (NAME, THESIS, REPORT, PAPER, PROPOSAL_DATE, PROPOSAL_APPROVED, TITLE, ORALS_DATE, PASS, FAIL, COND_PASS, MAJ_PROF, MEMBER_2, MEMBER_3)
 - VI. STUDENT_STATUS (NAME, STATUS_DATE, REG, PROV, SPEC, PROB)
 - VII. STUDENT_INFO (NAME, SSAN, LOCAL_ADDRESS, PERMANENT_ADDRESS, LOCAL_TEL_NO, SEX, RESIDENCY_DATE, CITIZENSHIP, PREVIOUS_NAME, GRAD_APP_DATE, GRE_OFFICIAL, GRE_VERBAL_SC, GRE_QUANT_SC, GRE_ANALYTICAL_SC, ADV_GRE_OFFICIAL, ADV_GRE_FLD, ADV_GRE_SC, TOEFL_OFFICIAL, TOEFL_SC, TOEFL_DATE, FINANCE_REQ, MEDICAL_REQ, LETTERS_OF_REC, CURRICULUM, ADVISOR, PROG_ENTRY_DATE, EXP_GRAD_DATE, PROG_OFF_STUDY_APP_DATE, CS_INFO_DOC_DATE, CMPSC_APP_DOC_DATE, PROC_FEES_PD)
 - VIII. TEACHING_ASSISTANT (NAME, CMPSC_200, CMPSC_201, CMPSC_202, CMPSC_203, CMPSC_204, CMPSC_205, CMPSC_206, CMPSC_300, CMPSC_305, CMPSC_306, CMPSC_405, CMPSC_410, CMPSC_420, CMPSC_430, CMPSC_450, CMPSC_560, CMPSC_580)
 - IX. UNIV_ATTENDED (NAME, DATES_ATTENDED, SCHOOL, SUBJECT_AREA, GPA, SCALE, DEGREE, DEGREE_DATE)
-

Figure 4.1
GSRDB Data Dictionary

Figure 4.1: Continued

Dictionary

Attribute	Relation	Field Length	Domain	Value	Constraint
1. ADV_GRE_OFFICIAL	VII	1	Alpha	Y or N	A
2. ADV_GRE_FIELD	VII	9	Alpha	All Alpha	
3. ADV_GRE_SC	VII	3	Numeric	All Positive Integer	
4. ADVISOR	VII	15	Alpha	All Alpha	
5. CRS_NO	I, III	9	Alpha/Numeric	All Alpha and Positive Integer	
6. CRS_NAME	I	20	Alpha	All Alpha	
7. CREDIT	I	1	Numeric	1, 2, 3, 4, 5, 6	
8. CURRENT_GPA	II	3	Numeric	All Positive Integer	
9. CRS_GRD	III	1	Alpha	A, B, C, D, F	B
10. COND_PASS	V	1	Alpha	X	
11. CITIZENSHIP	VII	10	Alpha	All Alpha	
12. CURRICULUM	VII	5	Alpha	CMPSC	
13. CS_INFO_DOC_DATE	VII	7	Alpha/Numeric	e.g., 17DEC82	
14. CMPSC_APP_DOC_DATE	VII	7	Alpha/Numeric	e.g., 3NOV82	
15. CMPSC_200	VIII	1	Alpha	X	
16. CMPSC_201	VIII	1	Alpha	X	

Figure 4.1: Continued.

Attribute	Relation	Field Length	Domain	Value	Constraint
17. CMPSC_202	VIII	1	Alpha	X	
18. CMPSC_203	VIII	1	Alpha	X	
19. CMPSC_204	VIII	1	Alpha	X	
20. CMPSC_205	VIII	1	Alpha	X	
21. CMPSC_206	VIII	1	Alpha	X	
22. CMPSC_300	VIII	1	Alpha	X	
23. CMPSC_305	VIII	1	Alpha	X	
24. CMPSC_306	VIII	1	Alpha	X	
25. CMPSC_405	VIII	1	Alpha	X	
26. CMPSC_410	VIII	1	Alpha	X	
27. CMPSC_420	VIII	1	Alpha	X	
28. CMPSC_430	VIII	1	Alpha	X	
29. CMPSC_450	VIII	1	Alpha	X	
30. CMPSC_560	VIII	1	Alpha	X	
31. CMPSC_580	VIII	1	Alpha	X	
32. DEF	III	1	Alpha	X	D
33. DATES_ATTENDED	IX	10	Alpha/Numeric	e.g., JUN79JUN80	

Figure 4.1: Continued.

Attribute	Relation	Field Length	Domain	Value	Constraint
34. DEGREE	IX	5	Alpha	MS, MA, BA, BS, PHD	E
35. DEGREE_DATE	IX	5	Alpha/Numeric	e.g., DEC82	F
36. EXP_GRAD_DATE	VII	5	Alpha/Numeric	e.g., MAY83	
37. FALL	IV	1	Alpha	X	G
38. FAIL	V	1	Alpha	X	C
39. FINANCE_REQ	VII	1	Alpha	Y, N	H
40. GRAD_APP_DATE	VII	7	Alpha/Numeric	e.g., 10SEP81	
41. GRE_OFFICIAL	VII	1	Alpha	Y, N	I
42. GRE_VERBAL_SC	VII	3	Numeric	All Positive Integer	
43. GRE_QUANT_SC	VII	3	Numeric	All Positive Integer	
44. GRE_ANALYTICAL_SC	VII	3	Numeric	All Positive Integer	
45. GPA	IX	3	Numeric	e.g., 375	
46. I_SPEC	IV	1	Alpha	X	J
47. LOCAL_ADDRESS	VII	47	Alpha/Numeric	e.g., 331FREMONTST MANHATTANKS 66502	
48. LOCAL_TEL_NO	VII	7	Numeric	e.g., 5372358	

Figure 4.1: Continued.

Attribute	Relation	Field Length	Domain	Value	Constraint
49. LETTERS_OF_REC	VII	1	Numeric	1, 2, 3	
50. MAJOR	III	1	Alpha	X	D
51. MS	IV	1	Alpha	X	J
52. MAJ_PROF	V	15	Alpha	All Alpha	
53. MEMBER_2	V	15	Alpha	All Alpha	
54. MEMBER_3	V	15	Alpha	All Alpha	
55. MEDICAL_REQ	VII	1	Alpha	Y, N	H
56. NAME	II, III, IV, V, VI, VII, VIII, IX	35	Alpha	All Alpha	
57. ORALS_DATE	V	5	Alpha/Numeric	e.g., NOV82	
58. PHD	IV	1	Alpha	X	J
59. PAPER	V	1	Alpha	X	K
60. PROPOSAL_DATE	V	5	Alpha/Numeric	e.g., AUG82	
61. PROPOSAL_APPROVED	V	1	Alpha	Y, N	
62. PASS	V	1	Alpha	X	C
63. PROV	VI	1	Alpha	X	L

Figure 4.1: Continued.

Attribute	Relation	Field Length	Domain	Value	Constraint
64. PROV	VI	1	Alpha	X	L
65. PERMANENT_ADDRESS	VII	47	Alpha/Numeric	e.g., 35GOULDSTREET CLIJTONNJ 07013	
66. PREVIOUS_NAME	VII	15	Alpha	All Alpha	
67. PROG_ENTRY_DATE	VII	5	Alpha/Numeric	e.g., SEP81	
68. PROG_OFF_STUDY_APP_DATE	VII	5	Alpha/Numeric	e.g., AUG82	
69. PROC_FEES_PD	VII	1	Alpha	Y, N	
70. REPEAT	III	1	Alpha	X	
71. REPORT	V	1	Alpha	X	K
72. REG	VI	1	Alpha	X	L
73. RESIDENCY_DATE	VII	5	Alpha/Numeric	e.g., JUN81	
74. SEM_YR	III	4	Alpha/Numeric	FA, SP, SU, (Plus YR) e.g., FA81	
75. SUPPORT	III	1	Alpha	X	D
76. SPRING	IV	1	Alpha	X	G
77. SUMMER	III	1	Alpha	X	G

Figure 4.1: Continued.

Attribute	Relation	Field Length	Domain	Value	Constraint
78. STATUS_DATE	VI	4	Alpha/Numeric	FA, SP, SU (Plus YR) e.g., SP80	
79. SPEC	VI	1	Alpha	X	L
80. SSAN	VII	9	Alpha/Numeric	All Positive Integer	
81. SEX	VII	1	Alpha	M, F	
82. SCHOOL	IX	25	Alpha	All Alpha	
83. SUBJECT_AREA	IX	5	Alpha	All Alpha	
84. SCALE	IX	4	Numeric	All Positive Integer	
85. THESIS	V	1	Alpha	X	K
86. TITLE	V	50	Alpha/Numeric	All Alpha and Positive Integer	
87. TOEFL_OFFICIAL	VII	1	Alpha	Y, N	M
88. TOEFL_SC	VII	3	Numeric	All Positive Integer	H
89. TOEFL_DATE	VII	5	Alpha/Numeric	e.g., JAN81	H
90. YEAR	IV	2	Numeric	All Positive Integer	

Constraint Notes

A - This attribute is related to attributes 2 and 3; data values should appear in both 2 and 3 when a value is entered for this attribute.

Figure 4.1: Constraint Notes Continued.

- B - This attribute is related to attributes 5, 6, 7 and 74; data values should appear in these attributes when a value is entered for this attribute. Additionally, this attribute is related to attributes 37, 50, 75; one of these attributes should also contain a value when this attribute receives a value.
- C - Attributes 10, 38, and 62 are mutually exclusive; therefore, only one of these attributes may possess a value in each tuple.
- D - Attributes 32, 50, and 75 are mutually exclusive; therefore, only one of these attributes may possess a value in each tuple.
- E - This attribute is related to attributes 33, 35, 82, and 83; data values should appear in these attributes when a value is entered for this attribute.
- F - This attribute is related to attributes 33, 34, 82, and 83; data values should appear in these attributes when a value is entered for this attribute.
- G - Attributes 37, 76, and 77 are mutually exclusive; therefore, only one of these attributes may possess a value in each tuple.
- H - Attributes 39, 55, 88, and 89 are related to attribute 11; data values for these attributes should appear when attribute 11's value $\neq$ U.S.A.
- I - This attribute is related to attributes 42, 43, and 44; data values should appear in all three attributes when a value is entered for this attribute.
- J - Attributes 46, 51, and 58 are mutually exclusive; therefore, only one of these attributes may possess a value in each tuple.
- K - Attributes 59, 71, and 85 are mutually exclusive; therefore, only one of these attributes may possess a value in each tuple.

Figure 4.1: Constraint Notes Continued.

- L -** Attributes 63, 64, 72, and 79 are mutually exclusive; therefore, only one of these attributes may possess a value in each tuple.
- M -** This attribute is related to attributes 88 and 89; data values should appear in both 88 and 89 when a value is entered for this attribute.

**Figure 4.1
GSRDB Data Dictionary**

10. Data volatility

11. INGRES limitations

Security, efficiency, flexibility, growth and performance issues were considered; however, they were not the major concerns. Suggestions for improving performance and data integrity are found in Section 5.0. With this in mind, my intent is to discuss each relation that has been designed versus the considerations listed above. Additionally, the 3NF relations designed by the ADBG system, Bernstein's algorithm #2, will be considered in this analysis.

The relation `STUDENT_INFO` was designed to provide an overall descriptive view of each student. This was accomplished by selecting most of the attributes for this relation from the initial application forms that the prospective student completes. The 31 attributes in this relation form the basic administrative and processing information for each student. The attributes `GRE_OFFICIAL`, `ADV_GRE_OFFICIAL`, `TOEFL_OFFICIAL`, `FINANCE_REQ`, `MEDICAL_REQ`, `LETTERS_OF_REC`, and `PROC_FEES_PD` provide the Department with information that is critical in monitoring both a U.S. and foreign applicant's progress in meeting the Graduate School and Department requirements. This relation supports the Department's data requirements, operational functions and control tasks during initial student processing and the Department's policy to have on hand and periodically review student administrative information. This relation supports the entire first screen designed for transaction processing, thereby reducing front-end programming requirements. User friendliness was supported by designing the attributes to reduce user input requirements. Eight of the 31 attributes have values that consist

of one character each. Integrity and consistency constraints were identified for 13 of the 31 attributes.

Constraints were identified for these attributes because of the significance of their data values to the Department (see Figure 4.1, footnotes A, H, I and M). Concerning the volatility of data, update and deletion of specific attributes is minimal. Therefore, the majority of times this relation is accessed will be to review an applicant/student's progress in meeting processing requirements and for periodic reviews of student records.

The ADBG system merged this relation with four other relations with the same key. However, in order to support the transaction processing system design and based on significant differences in the data volatility and retrieval frequency of these relations, STUDENT_INFO must remain as a separate relation. This relation is in 3NF.

The relation, INITIAL_ADM_INFO was designed to contain the data to identify prospective Computer Science Department students. This relation contains data that is vital to forecast student population. The values for this relation are obtained from the Graduate School Application Form and this single relation consisting of 8 attributes enables the Department to identify prospective students. Additionally, this relation supports the Computer Science Department's efforts to determine a student's educational goal. Integrity and consistency constraints were identified for 6 of the 8 attributes of this relation. These constraints are necessary because the values of these attributes are mutually exclusive (see Figure 4.1, footnotes G and J).

Concerning the volatility of data, update and deletion actions on this relation will be minimal. Basically, once the relation is created

it should not be necessary to change the value of any attribute. However, the Department can retrieve this relation and perform efficient queries to forecast future student population.

User friendly transaction processing was supported by designing the attributes to reduce user input requirements. Six of the 8 attributes have values that consist of one character each. Front-end programming requirements are minimized, in that, the relation design supports the transaction processing system screen format.

The ADBG system merged this relation with four other relations with the same key. However, in order to support the transaction processing system design and based on significant differences in the data volatility and retrieval frequency of these relations, INITIAL_ADM_INFO must remain as a separate relation. This relation is in 3NF.

The MASTERS_RESEARCH relation was designed to contain all the data requirements associated with a master's student research program. This relation supports the Computer Science Department's operational functions, control and planning requirements and operating policies concerning the Department's overall responsibilities in this area. For instance, this relation provides the Department with the ability to perform timely student advising, record proposal data and the results of oral examinations, and insure that a major professor and committee have been selected by a student in a timely manner.

This relation minimizes front-end programming requirements, in that, it is designed to support the transaction processing system screen format.

User friendly transaction processing was supported by designing the attributes to reduce user input requirements. Six of the 14 attributes have values that consist of one character each.

Concerning data volatility, the Department will require accessing the relation approximately 4 times for update during the average student's assignment. Integrity and consistency constraints have been identified for 6 of the 14 attributes. These constraints are necessary because the values of these attributes are mutually exclusive (see Figure 4.1, footnotes C and K).

This relation was one of the five that the ADBG system merged together. However, in order to support the transaction processing system design and based on significant differences in data volatility and retrieval frequency of these relations, MASTERS_RESEARCH must remain as a separate relation. This relation is in 3NF.

The GRAD_COURSES and COURSE_LIST relations were designed to contain all relevant data concerning a student's program of study and additional course requirements, e.g., deficiency courses. The design of the COURSE_LIST relation adds efficiency and flexibility to the data base by containing the relevant information about all the courses that are taught by the Department. These relations contain the data requirements and support the operational functions and operational control responsibilities of the Department by containing the data that provide all the course information for a student, e.g., course number, course name, when the course was taken, if it has been repeated, course grade, course credit, and whether the course was a major, supporting, or deficiency course. These relations go beyond the scope of the program of study and provide vital information for the Department, faculty, and

student. The data in the GRAD_COURSE relation is perhaps the most volatile in the data base. Program of study changes should be seldom but grades must be updated each semester. The COURSE_LIST relation reduces input requirements by containing a tuple for each course the Department teaches. This eliminates the need to update the course name and credit hours when the program of study changes. These relations reduce front-end programming requirements, in that, they are designed to match the transaction processing system screens. User friendly transaction processing was supported by designing the attributes to reduce user input requirements. Four of the 8 attributes of the GRAD_COURSE relation have attribute values that consist of one character each.

Integrity and consistency constraints have been identified for 4 of the 8 attributes of the GRAD_COURSE relation. Constraints were identified based on the significance of the data values and because the value of certain attributes are mutually exclusive (see Figure 4.1, footnotes B and D). GRAD_COURSE and COURSE_LIST were unchanged by the ADBG system. They are in 3NF.

The relation GPA was designed to contain the data to provide a student's current grade point average. This relation provides the data to support the Department's information, control and operating policies. The data contained in this relation will be updated approximately once per semester. This relation minimizes front-end programming requirements, in that, it is designed to support the transaction processing system screen format.

The ADBG system merged this relation with four other relations with the same key. However, in order to support the transaction processing

system design and based on significant differences in data volatility, and retrieval frequency of the relations, GPA must remain as a separate relation. This relation is in 3NF.

The TEACHING_ASSISTANT relation was designed to identify students who feel they have the potential to become teaching assistants for particular courses. This relation supports Department information, control and planning tasks, and operating functions and policies.

User friendliness was supported by designing the attributes to reduce user input requirements. Additionally, the data contained in this relation is not very volatile and should be seldom updated. Retrieval should be limited to one time per semester. This relation minimizes front-end programming requirements, in that, its design supports the transaction processing system screen format.

The ADBG system merged this relation with 4 other relations with the same key. However, in order to support the transaction processing system design and based on significant differences in data volatility and retrieval frequency of these relations, TEACHING_ASSISTANT must remain as a separate relation. This relation is in 3NF.

The STUDENT_STATUS relation was designed to provide an audit trail of a student's status changes. This relation supports the Department's information requirements, control tasks, and operating policies.

User friendly transaction processing was supported by designing the attributes to reduce user input requirements. Four of the 6 attributes have values that consist of one character each. Integrity and consistency constraints have been identified for 4 of the 6 attributes. These constraints are necessary because the values of these attributes are mutually exclusive (see Figure 4.1, footnote L). This relation

minimizes front-end programming requirements, in that, its design supports the transaction processing system screen format. The data in this relation should not require frequent updates. Retrieval, for review purposes, should be no more than once per semester. This relation was not changed by the ADBG system. It is in 3NF.

The UNIV_ATTENDED relation was designed to contain the data concerning each student's educational background. This relation supports the Department's information requirements and operating policies. User friendly transaction processing was supported by designing the attributes to reduce user input requirements. This relation minimizes front-end programming requirements, in that, its design supports the transaction processing system screen format. The data in this relation should not require update after it is initially established. Retrieval, for review purposes, should occur approximately twice annually. Integrity and consistency constraints have been identified for 2 of the 8 attributes. These constraints are necessary due to the significance of the data of these attributes (see Figure 4.1, footnotes E and F). This relation was not changed by the ADBG system. It is in 3NF.

CHAPTER V

FUTURE ENHANCEMENT AND SUMMARY

5.0 Future Enhancements5.0.1 Design

Enhancements necessary to the data base design will become evident only after implementing a prototype system and performance criteria is established and evaluated. At this point, the schema can be evaluated based on quantitative information and performance measures. Performance measures, such as logical record access counts, total bytes transferred to satisfy an application, and total bytes in the data base, attempt to predict, as close as possible, physical data base performance in terms of elapsed time and physical storage space [11]. Therefore, the schema may be further refined. However, the major constraints of providing a user friendly transaction processing system may prevent refinement of the schema.

5.0.2 Security

Since most DBMS appear to an operating system as an application program, a sophisticated user can bypass this application program and go directly to the operating system to break any security measures that might be established. Since there is minimal security on this system, this area should be one of the first areas enhanced.

5.0.3 Data Dictionary

The data dictionary that is described in Section 4.1 could be enhanced to include the following:

1. Passwords (when established for security)
2. Listing of report names and all attributes used in each report
3. Listing of user names and all source document names controlled or received by each user
4. Listing of user names and all report names controlled or received by each user

5.0.4 Consistency

The values of two or more particular fields may be related and may have to observe some constraints (e.g., those described in Figure 4.1). An enhanced edit program to enforce these constraints would be a beneficial improvement to the system.

5.1 Summary

The impetus for this report was the Computer Science Department's need for an efficient, user friendly, responsive and manageable automated system to provide pertinent information to perform all applicant/student related administrative tasks. In order to accomplish this, a data base had to be designed that could accommodate all the data requirements of the Department. This could only be accomplished if all of the objectives listed in Section 1.3 were met. I have reviewed those objectives and am confident that they have been met.

In Chapter II, I discussed the basic challenge that all data base designers face, the challenge of arriving at a logical data base design that represents the real world. This is a complicated process because the designer has to consider not only how to model the real world, but also the limitations of the data base system which will implement it.

A method to abstract this process called the Entity-Relationship (E-R) approach was used. The E-R approach to logical data base design abstracts the design process by defining two major phases: (1) defining the enterprise schema using the E-R diagram, and (2) translating the enterprise schema into a user schema. The advantages of this approach were presented in Section 2.3

The objectives and methodology of the data base design were presented in Section 2.4. The design of the GSRDB followed a practical methodology consisting of:

1. Requirements Formulation and Analysis
2. Conceptual/Logical Design (E-R Diagrams)
3. Conceptual/Logical Design (Normalized Relations via the Automatic Data Base Generator system)

The fourth step of this methodology, implementation design (translation from Normalized Relations to the data base schema), was modified based on the following:

- a. The Computer Science Department currently has the 2.8 version of INGRES but is having a problem translating INGRES from its 16-bit word format to the 3220 32-bit word format.
- b. The Department will receive the 4.1 version of INGRES in mid-January 1983.
- c. Manuals for the 4.1 version of INGRES were not available.

The three facts listed above prevented the translation from the Normalized Relations to the INGRES data base schema.

Since the schema could not be presented in INGRES, the schema was presented using the intention form of the relations.

The next two major sections explained and presented the Requirements Formulation and Analysis step (Section 2.5) and the Conceptual Design step (Section 2.6). Subsections of 2.6 explained how

entities, relationships, attributes, values and key attributes were selected for the GSRDB.

Data base systems and models were reviewed in Chapter III. The relational model, including its advantages and challenges were discussed. Additionally, a comprehensive discussion of normalization and functional dependency is found in Section 3.2.

The ADBG system was used to determine if the relations that were designed were indeed in 3NF. A description of the ADBG system and its results concerning the designed relations was presented in Sections 3.3 and 3.4, respectively.

The schema, in intention form, data dictionary and the major design considerations were presented in Chapter IV. The data dictionary cross-referenced each attribute by relation and listed the field length, domain, value, and constraints of the values of the attributes in the GSRDB. Each relation of the schema for GSRDB was then analyzed based on the following design considerations:

1. Computer Science Department data requirements
2. Computer Science Department operational functions
3. Computer Science Department control and planning tasks
4. Computer Science Department operating policies
5. Support of a user friendly transaction processing system
6. Minimize the front-end programming requirements (the number of data manipulation commands required to move the data in and out of the data base)
7. User accessing requirements (ad-hoc queries)
8. Logical association of attributes (E-R diagrams)
9. Integrity and consistency

10. Data volatility

11. INGRES Limitations

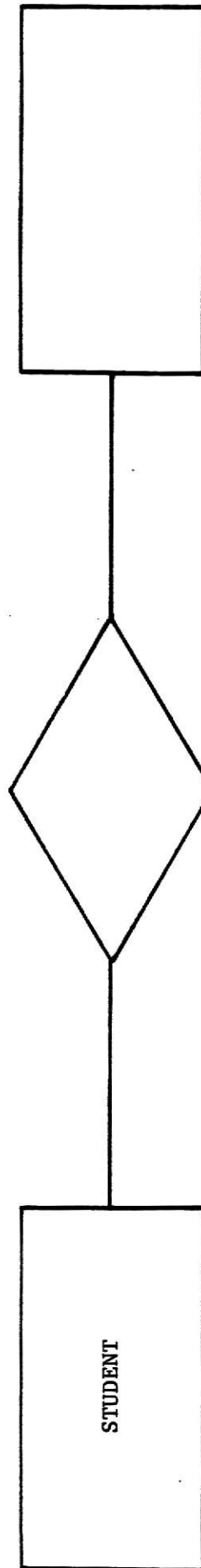
The results of this analysis is found in Section 4.2.

A discussion concerning future enhancements to the designed data base was presented in Chapter V.

BIBLIOGRAPHY

1. Bernstein, Philip A., "Synthesizing Third Normal Form Relations from Functional Dependencies," *ACM Transactions on Database Systems*, Vol. 1, No. 4, December 1976, pp. 277-298.
2. Cardenas, Alfonso F., Data Base Management Systems, Allyn and Bacon, Inc., 470 Atlantic Avenue, Boston, Massachusetts 02210, 1979, pp. 519.
3. Chen, Peter, Pin-Shan, "The Entity-Relationship Model: A Basis for the Enterprise View of Data," *Proc. AFIPS 1977 NCC*, Vol. 46, AFIPS Press, N.J., pp. 77-84.
4. Chen, Peter, Pin-Shan, "The Entity-Relationship Model: Toward a Unified View of Data," *ACM Transactions on Database Systems*, Vol. 1, No. 1, March 1976, pp. 9-36.
5. Chen, Peter, Pin-Shan, "The Entity-Relationship Approach to Logical Data Base Design," *The Q.E.D. Monograph Series*, Q.E.D. Information Sciences, Inc., 141 Linden Street, Wellesley, Massachusetts 02181, pp. 73.
6. Date, C. J., An Introduction to Database Systems, 3rd ed., Addison-Wesley Publishing Company, Reading, Massachusetts, 1981, pp. 574.
7. Epstein, Robert, "Creating and Maintaining a Database Using INGRES," Memorandum No. ERL-M77-71, December 16, 1977, University of California, Berkeley 94720, pp. 25.
8. Epstein, Robert, "A Tutorial on INGRES," Memorandum No. ERL-M77-25, December 15, 1977, (revised), University of California, Berkeley 94720, pp. 31.
9. Fisher, Paul S., Automatic Data Base Generator, Final Report, Computer and Information Sciences, 1116 Laramie, Manhattan, Kansas 66502, pp. 77.
10. Fisher, Paul S., "Database Design Technique," *Computer Communications*, Vol. 4, No. 6, December 1981, pp. 273-280.
11. Teorey, Toby, J., and Fry, James, P., Design of Database Structures, Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, 1982, pp. 492.
12. Woodfill, J.; Whyte, M.; Ubell, M.; Siegle, P.; Ries, D.; Meyer, M.; Hawthorn, P.; Epstein, B.; Berman, R.; and Allman, E., INGRES, Version 6.3, Reference Manual, April 2, 1981.

APPENDIX A
INITIAL ENTITY DIAGRAMS WITH RELATIONSHIPS,
ATTRIBUTES AND VALUES



The following list of attributes were in the initial design of the STUDENT ENTITY:

ADV_GRE_OFFICIAL	FINANCE_REQ	LOCAL_TEL_NO	RESIDENCY_DATE
ADV_GRE_FIELD	GRAD_APP_DATE	LETTERS_OF_REC	SEX
ADV_GRE_SC	GRE_OFFICIAL	MEDICAL_REC	* STU_NAME
CITIZENSHIP	GRE_VERBAL_SC	PERMANENT_ADDRESS	TOEFL_DATE
CS_INFO_DOC_DATE	GRE_QUANT_SC	PERVIOUS_NAME	TOEFL_OFFICIAL
CMPSA_APP_DOC_DATE	GRE_ANALYTICAL_SC	PROG_ENTRY_DATE	TOEFL_SC
EXP_GRAD_DATE	LOCAL_ADDRESS	PROC_FEES_PD	

These attributes will not be duplicated on the other diagrams in this Appendix.

Figure A.1
Student Entity

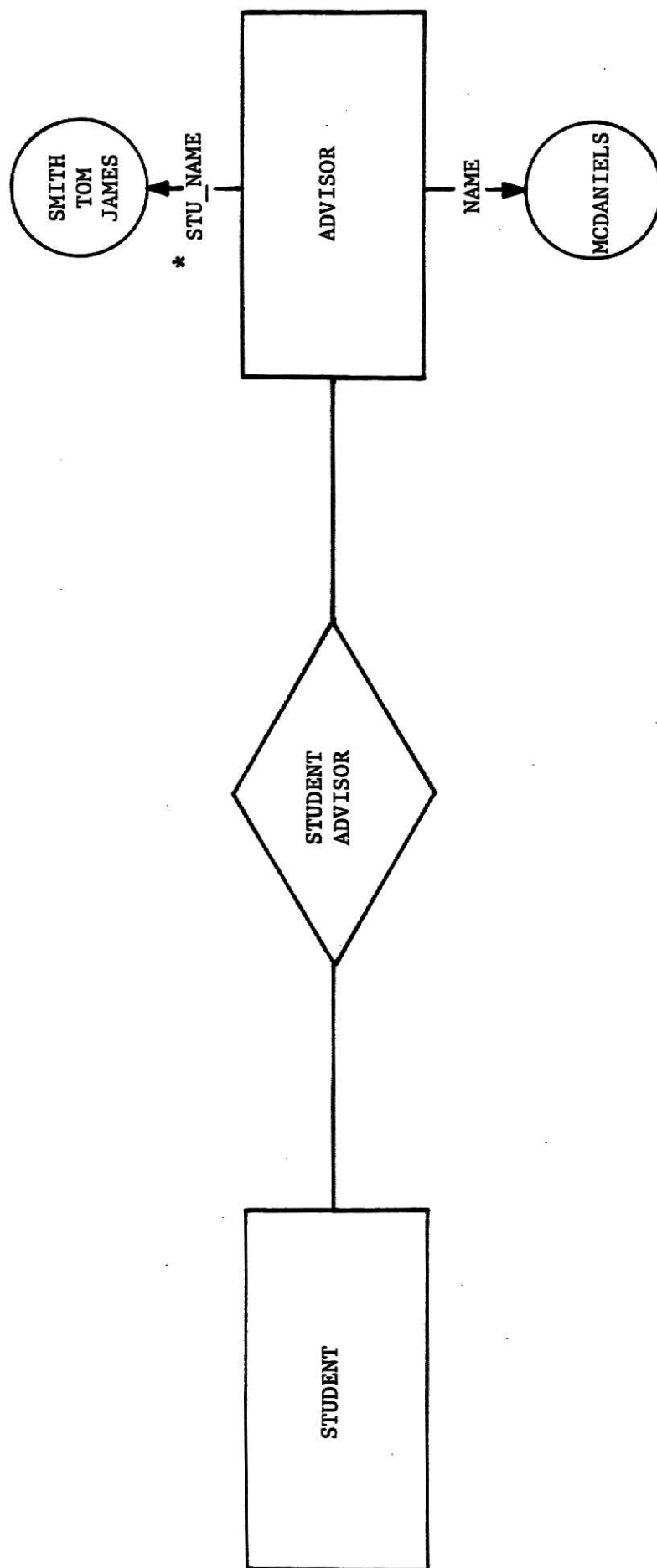


Figure A.2
STUDENT ADVISOR Relationship

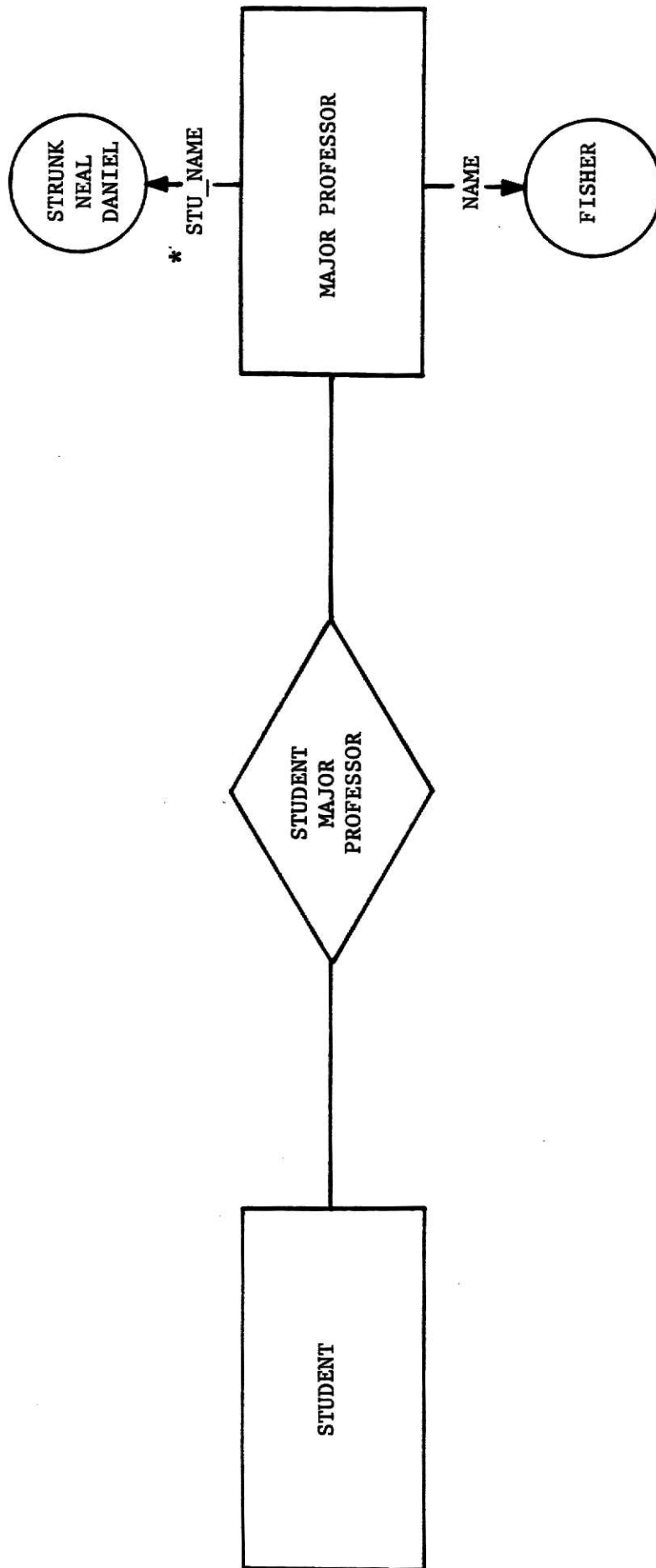


Figure A.3
STUDENT MAJOR PROFESSOR Relationship

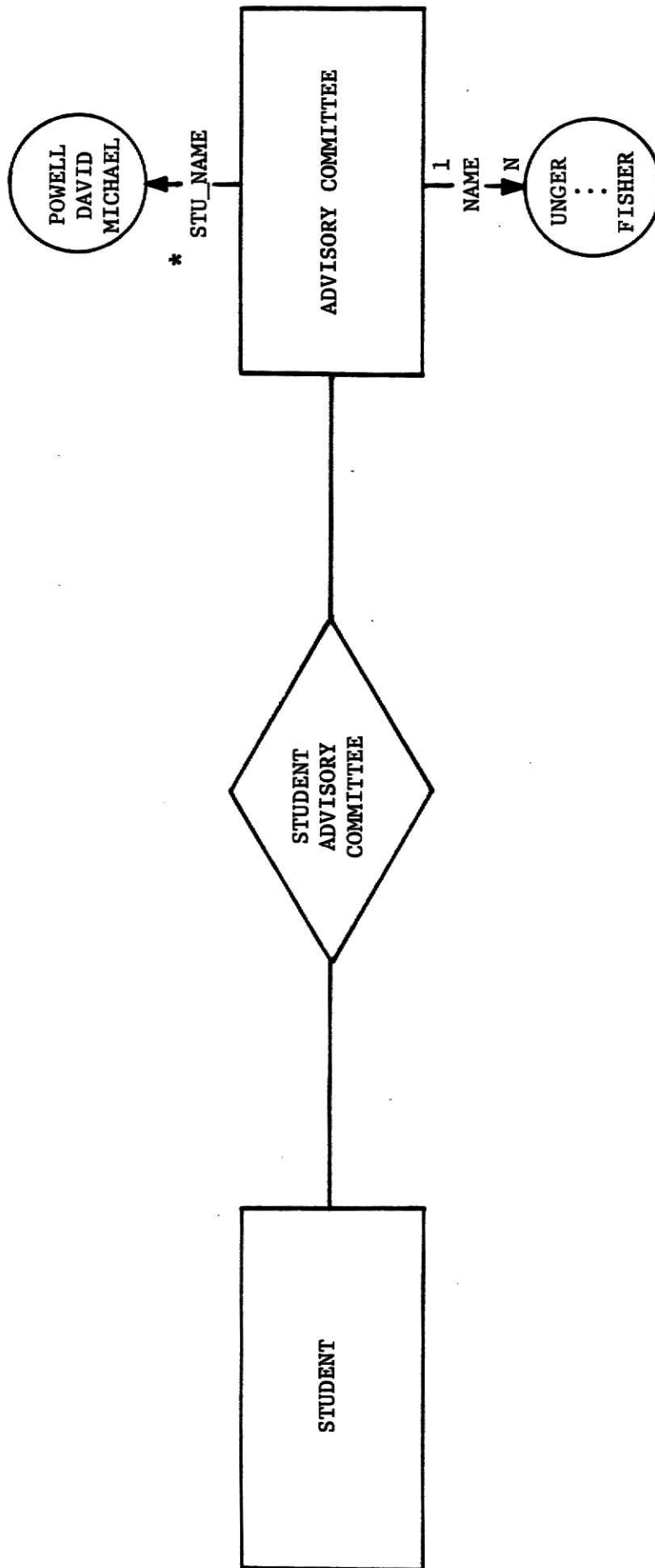


Figure A.4
STUDENT_ADVISORY COMMITTEE Relationship

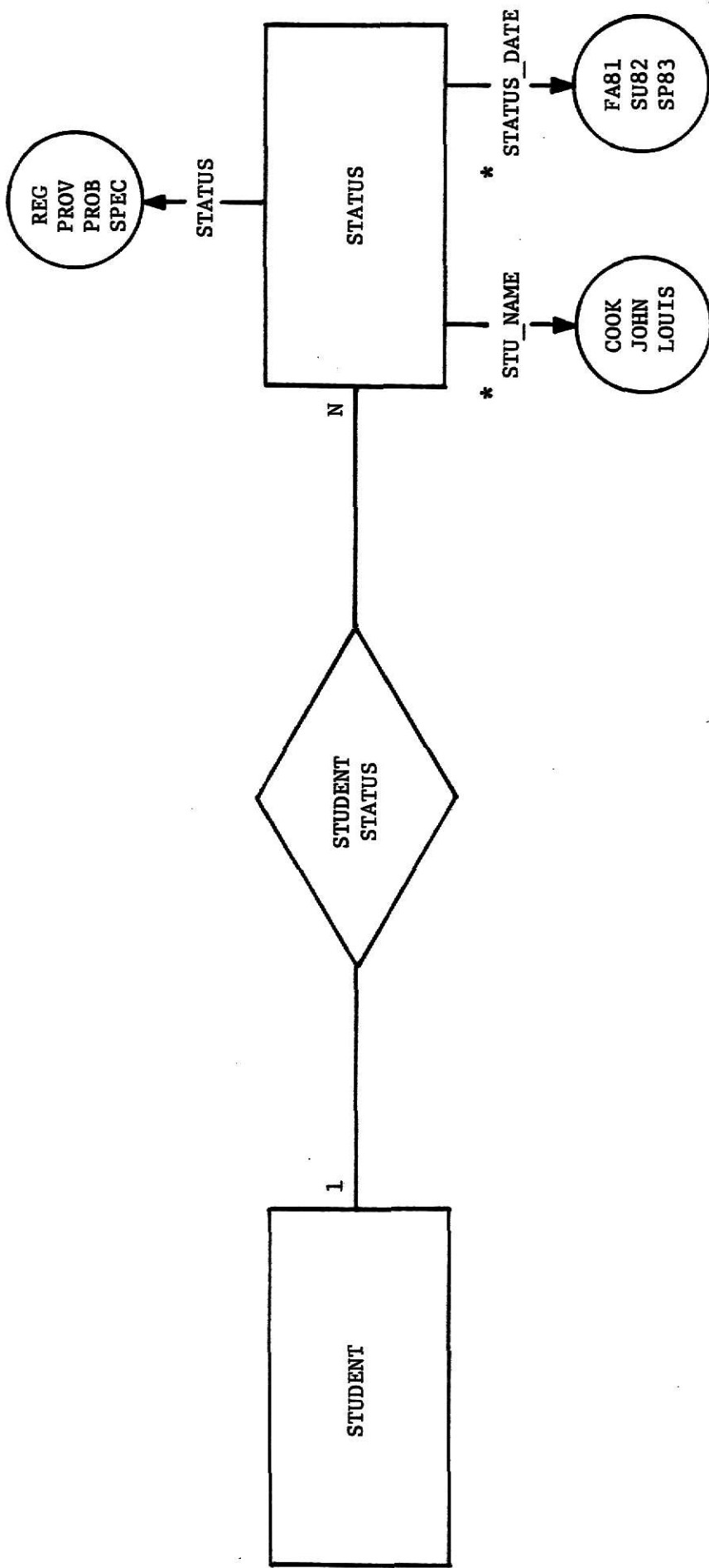


Figure A.5
STUDENT STATUS Relationship

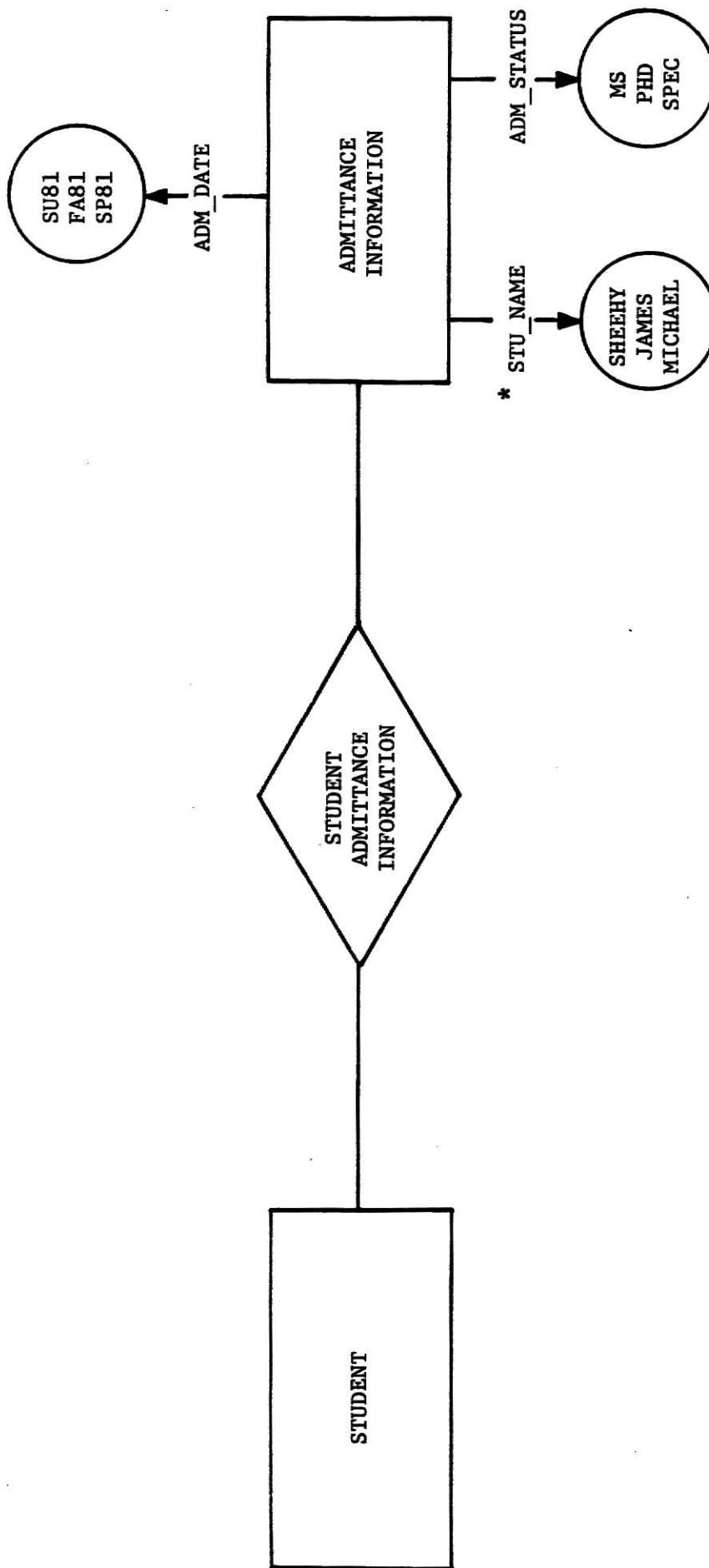


Figure A.6
STUDENT_ADMITTANCE INFORMATION Relationship

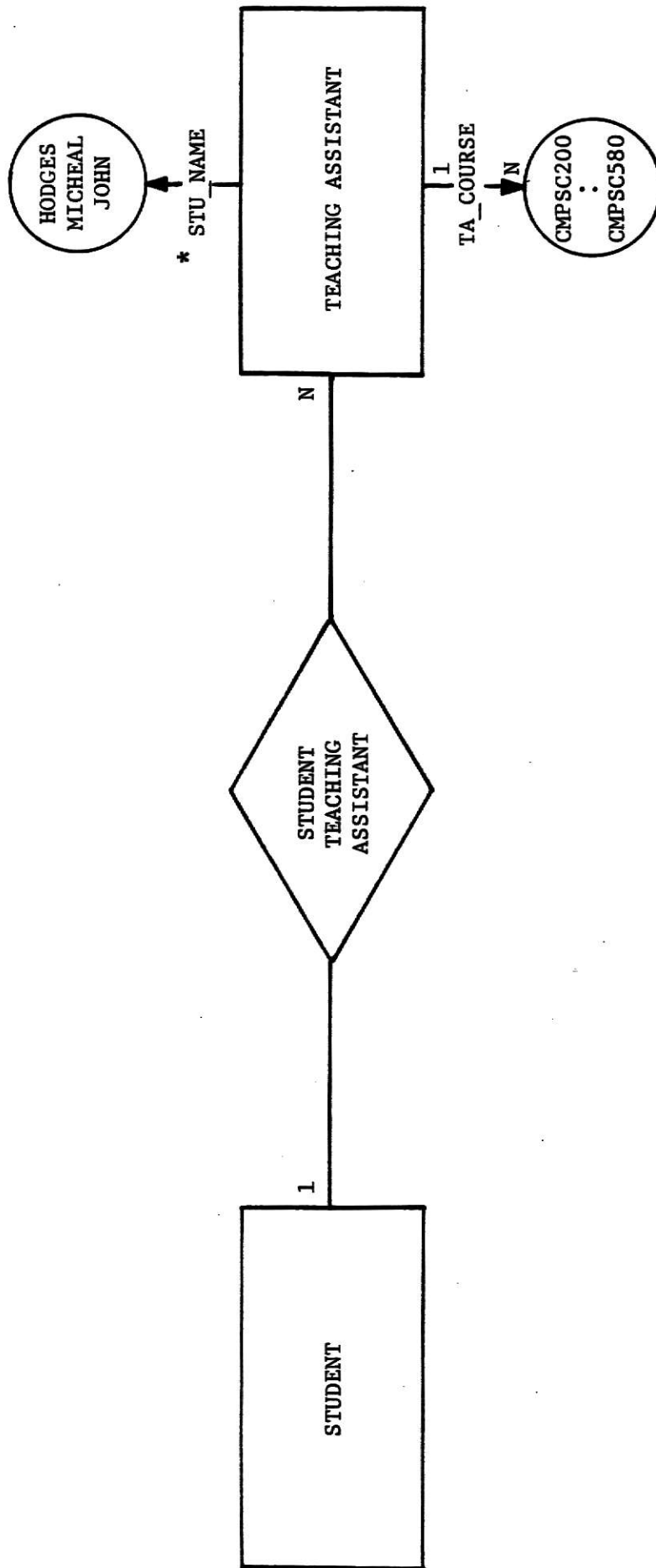


Figure A.7
STUDENT_TEACHING ASSISTANT Relationship

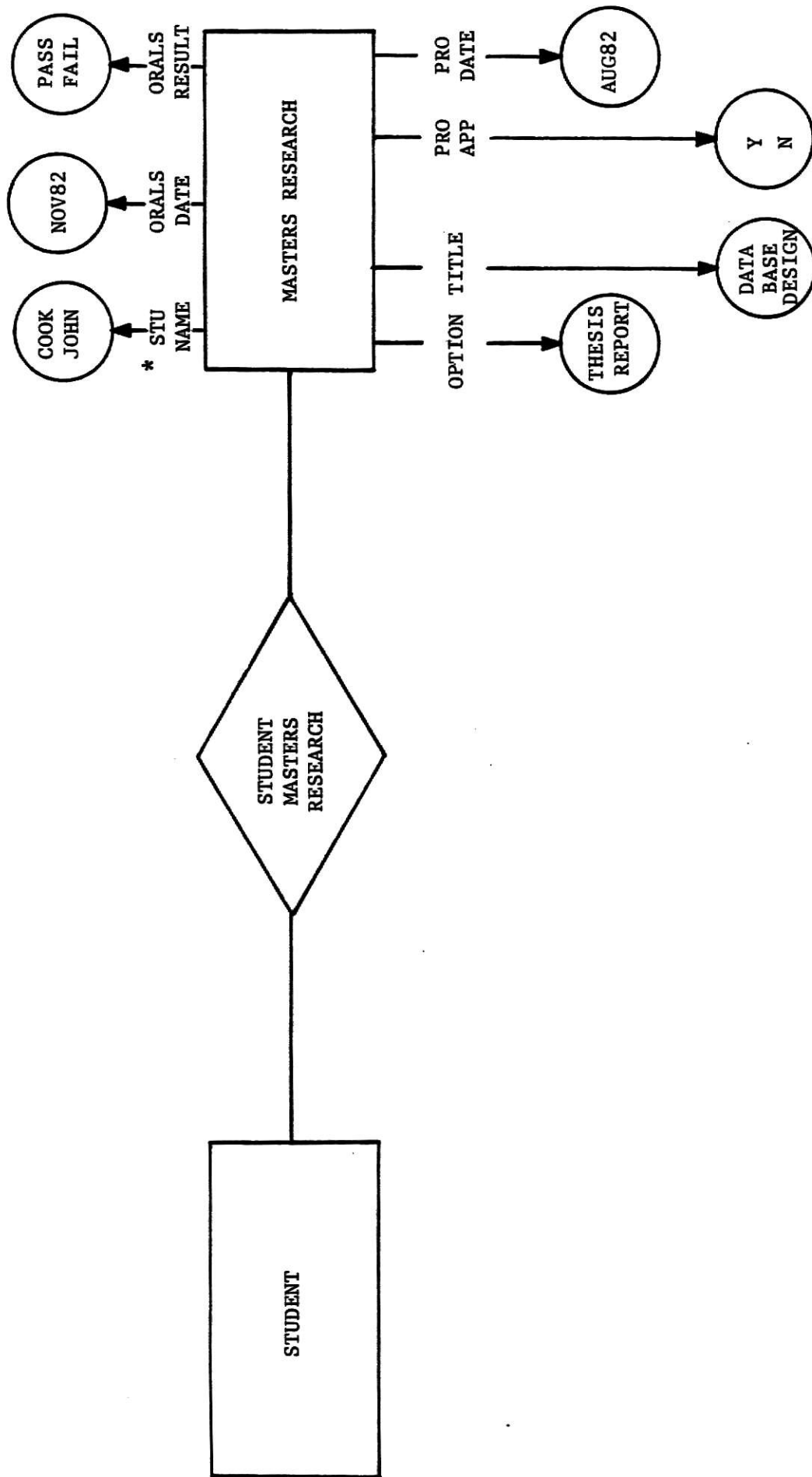
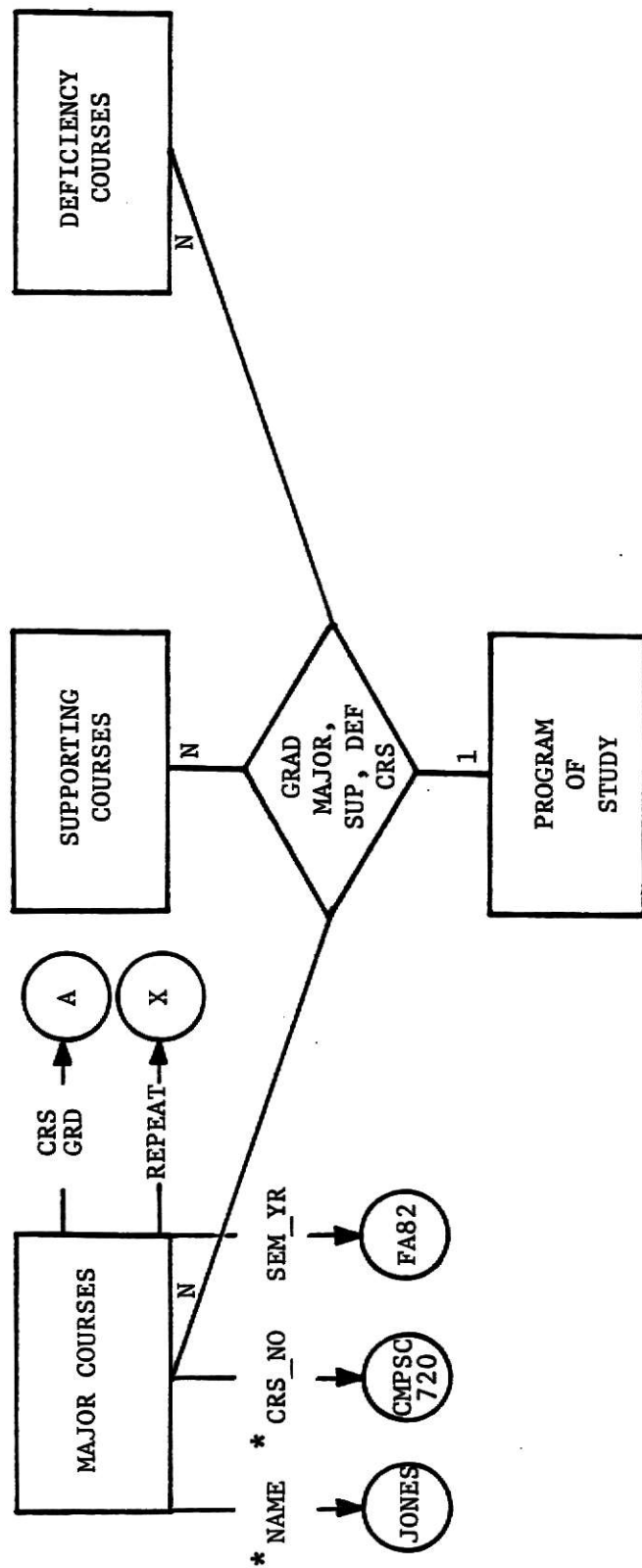


Figure A.8
STUDENT_MASTERS RESEARCH Relationship



The attributes of the entities MAJOR COURSE, SUPPORTING COURSE, and DEFICIENCY COURSE are identical. Therefore, these attributes are shown only one time. The attributes of the entity PROGRAM OF STUDY are shown in Figure A.12.

Figure A.9
MAJOR, SUPPORTING, DEFICIENCY COURSE PROGRAM OF STUDY Relationship

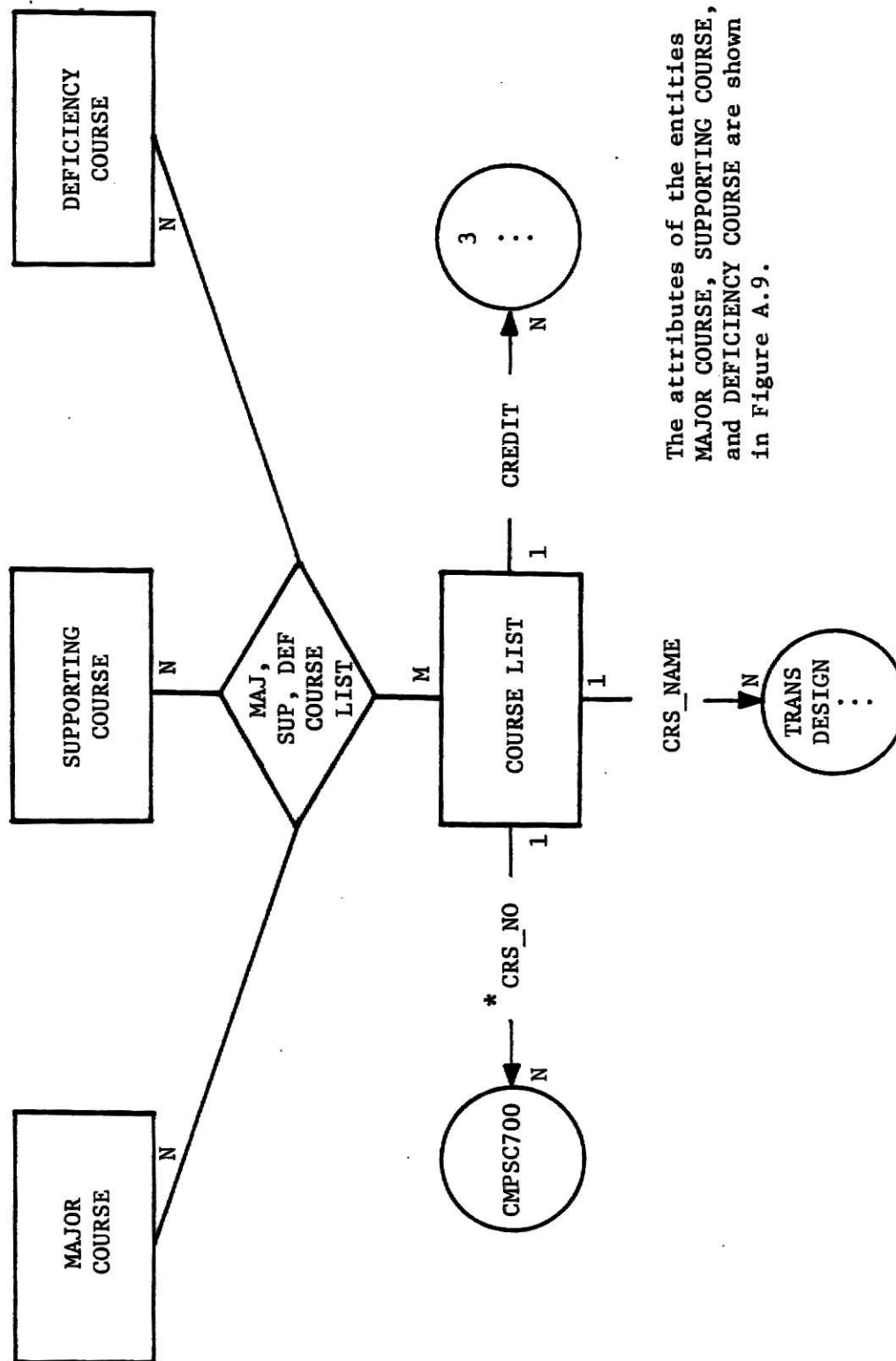


Figure A.10
MAJOR, SUPPORTING, DEFICIENCY COURSE LIST Relationship

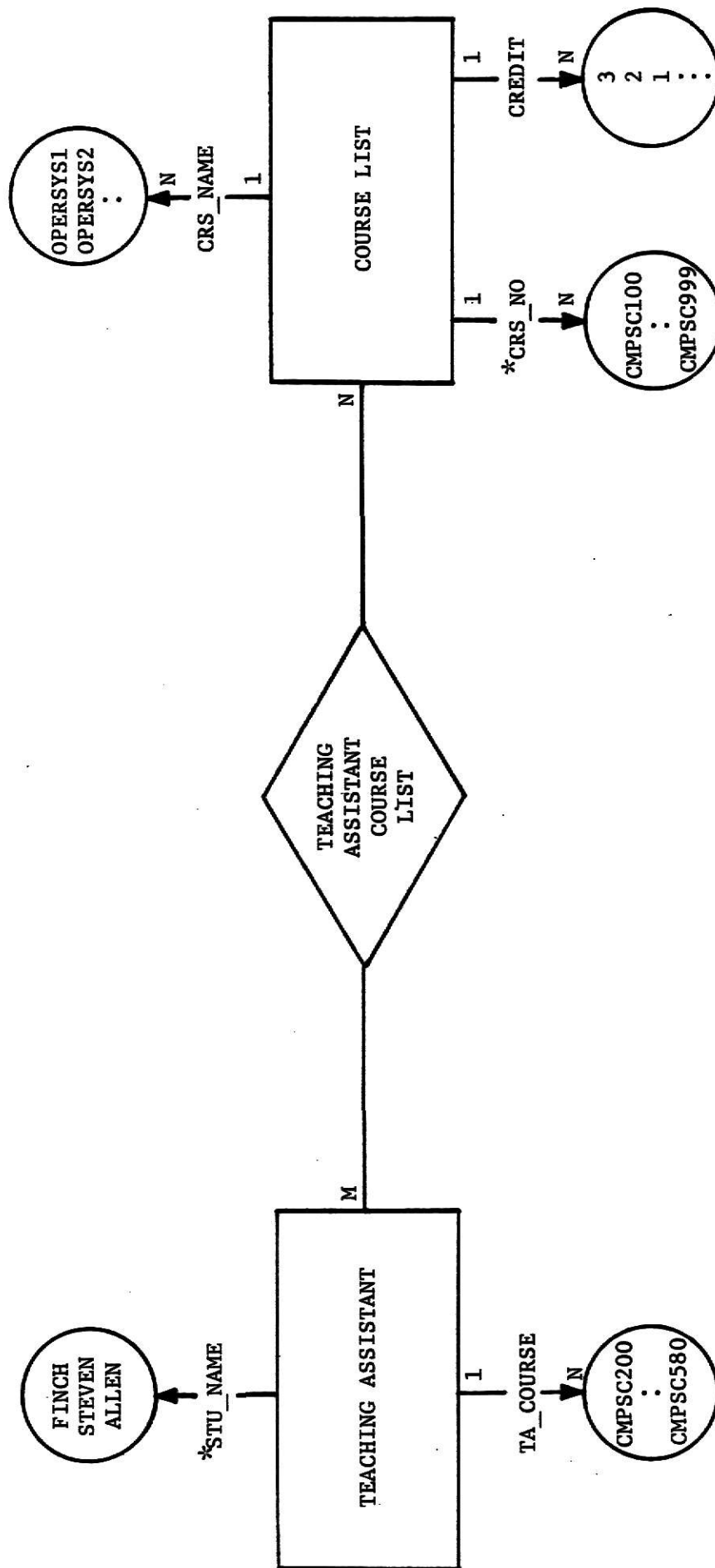


Figure A.11
TEACHING ASSISTANT COURSE LIST Relationship

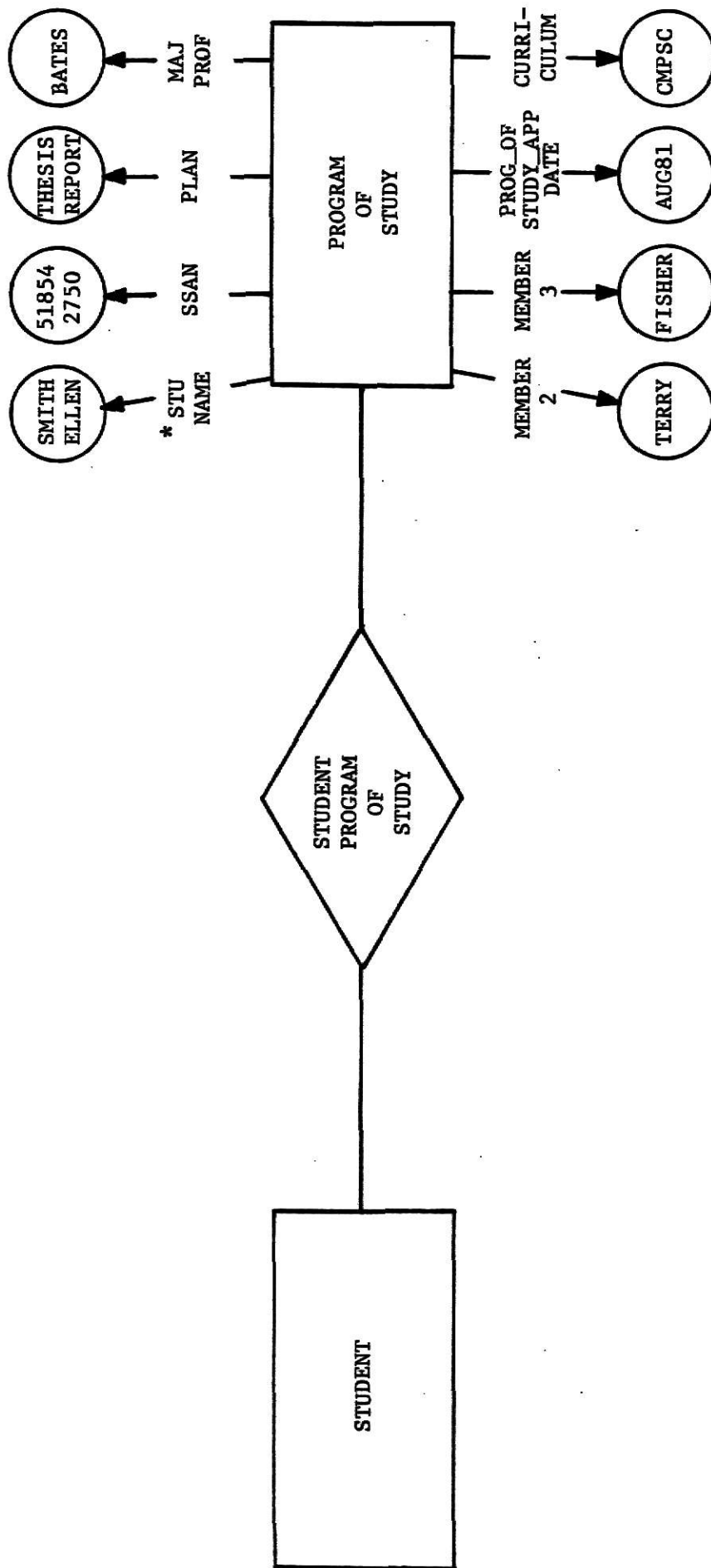


Figure A.12
STUDENT_PROGRAM OF STUDY Relationship

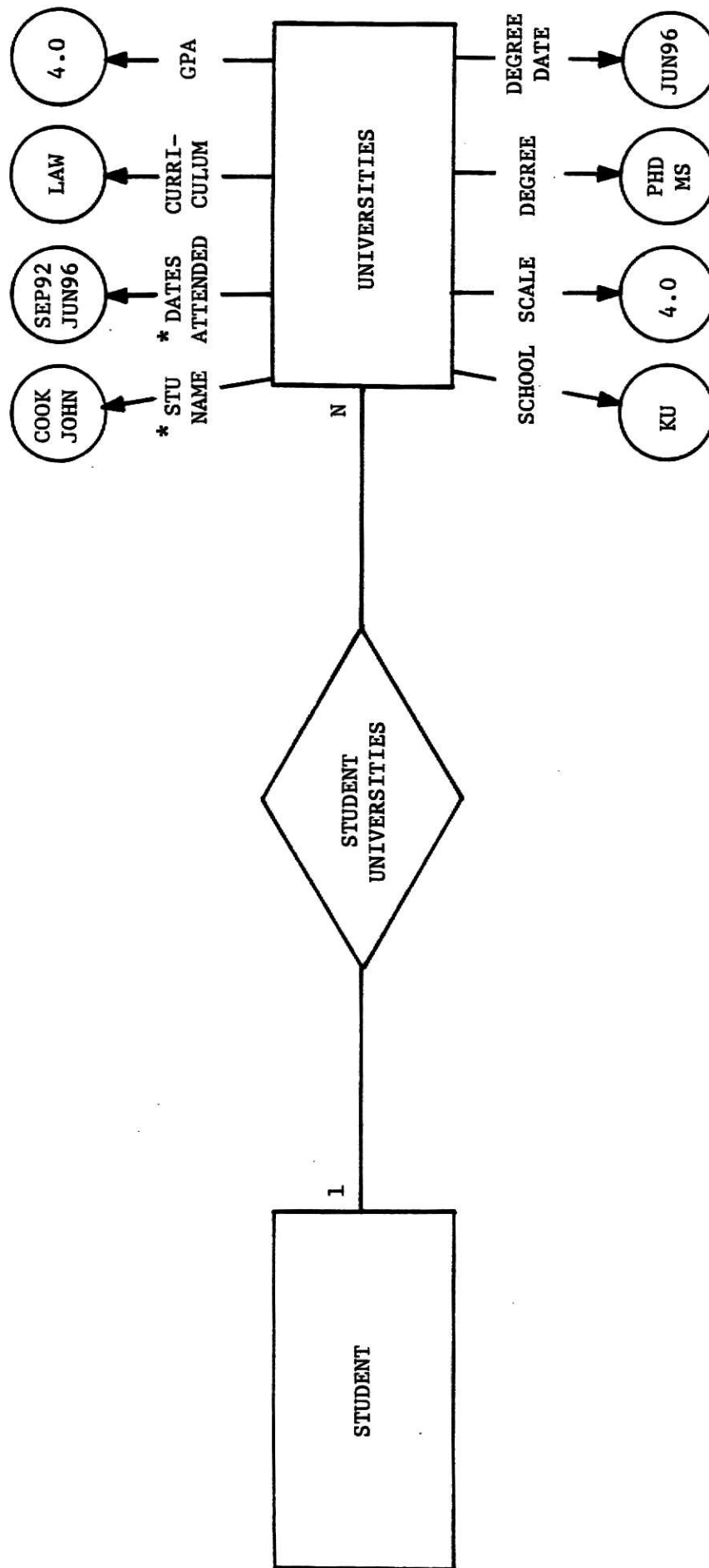


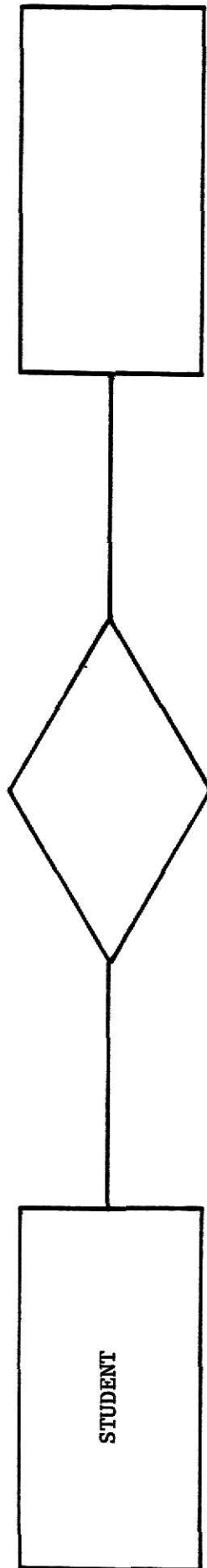
Figure A.13
STUDENT_UNIVERSITIES Relationship

APPENDIX B

**FINAL ENTITY DIAGRAMS WITH RELATIONSHIPS,
ATTRIBUTES AND VALUES**

Attributes added to STUDENT ENTITY based on design change:

ADVISOR
 PROG_OF_STUDY_APP_DATE
 SSAN
 CURRICULUM



The following list of attributes were in the initial design of the STUDENT ENTITY:

ADV_GRE_OFFICIAL	FINANCE_REQ	LOCAL_TEL_NO	RESIDENCY_DATE
ADV_GRE_FIELD	GRAD_APP_DATE	LETTERS_OF_REC	SEX
ADV_GRE_SC	GRE_OFFICIAL	MEDICAL_REC	* STU_NAME
CITIZENSHIP	GRE_VERBAL_SC	PERMANENT_ADDRESS	TOEFL_DATE
CS_INFO_DOC_DATE	GRE_QUANT_SC	PREVIOUS_NAME	TOEFL_OFFICIAL
CMPSA_APP_DOC_DATE	GRE_ANALYTICAL_SC	PROG_ENTRY_DATE	TOEFL_SC
EXP_GRAD_DATE	LOCAL_ADDRESS	PROC_FEES_PD	

Figure B.1
 Student Entity

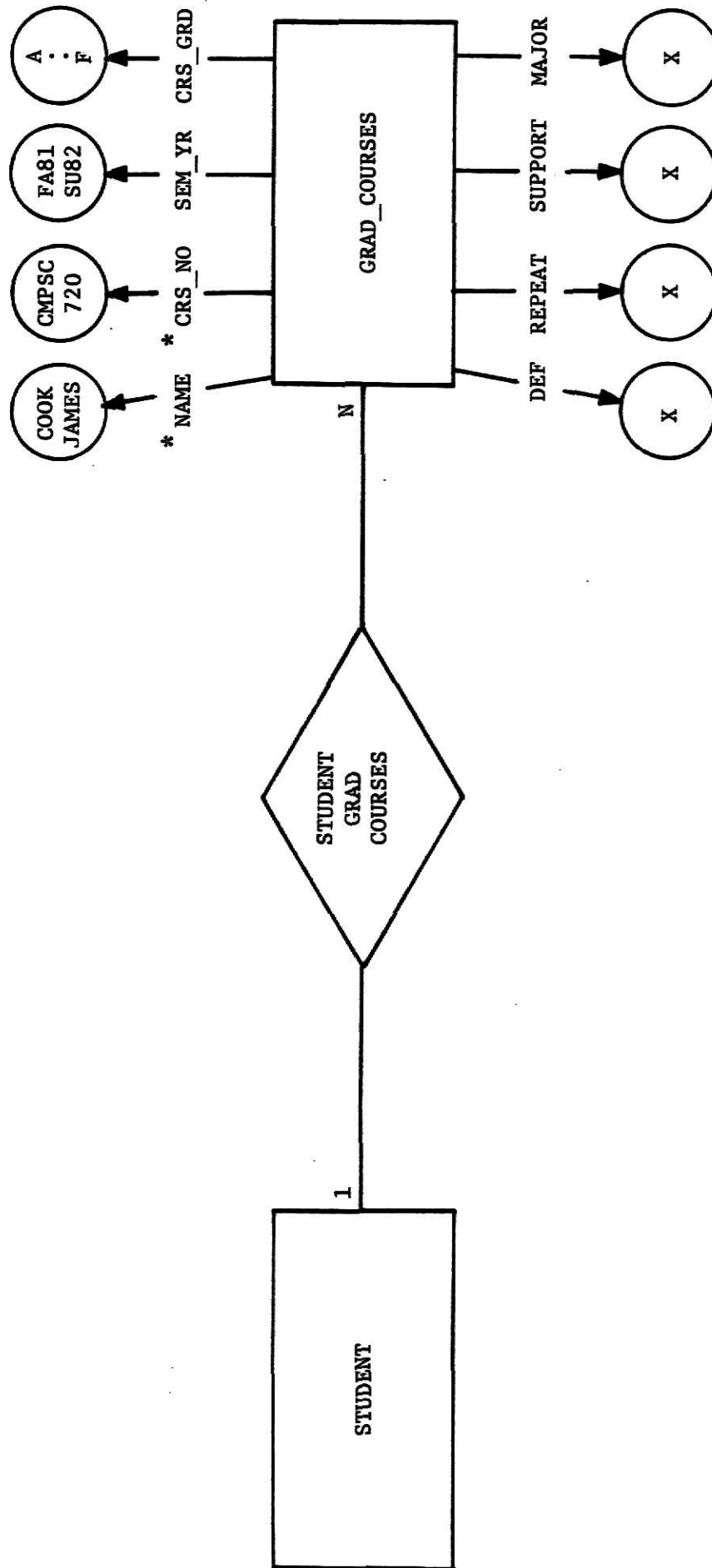


Figure B.2
STUDENT_GRAD_COURSES Relationship

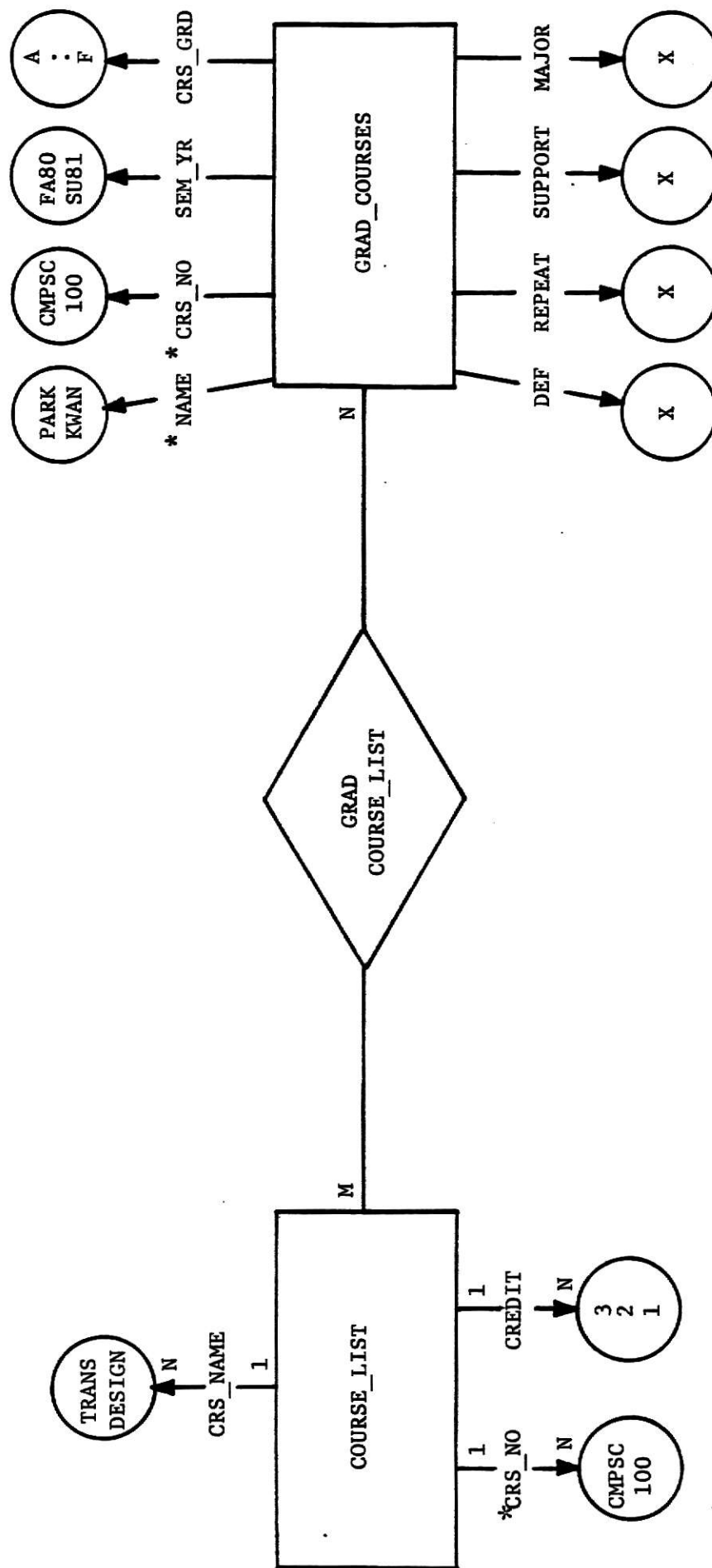


Figure B.3
COURSE LIST GRAD COURSES Relationship

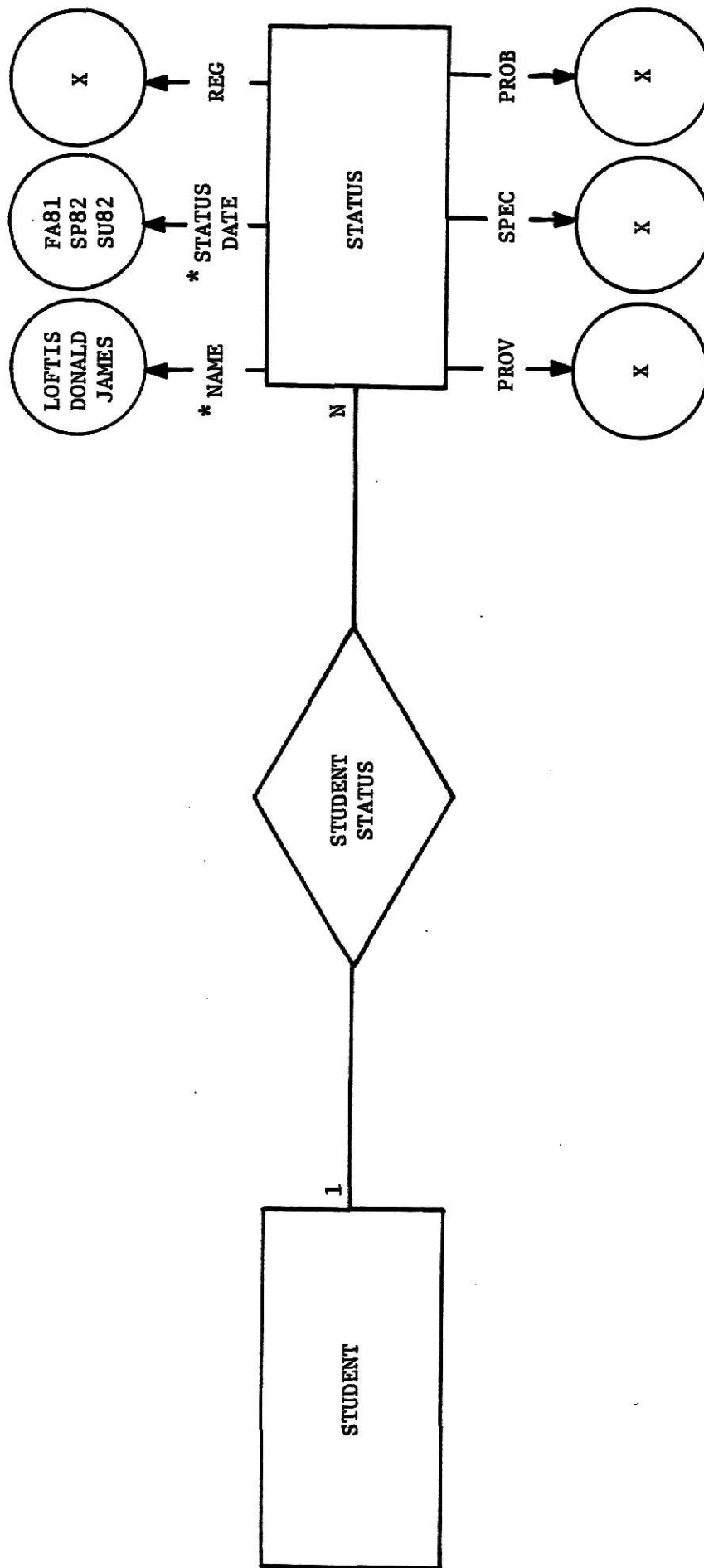


Figure B.4
STUDENT_STATUS Relationship

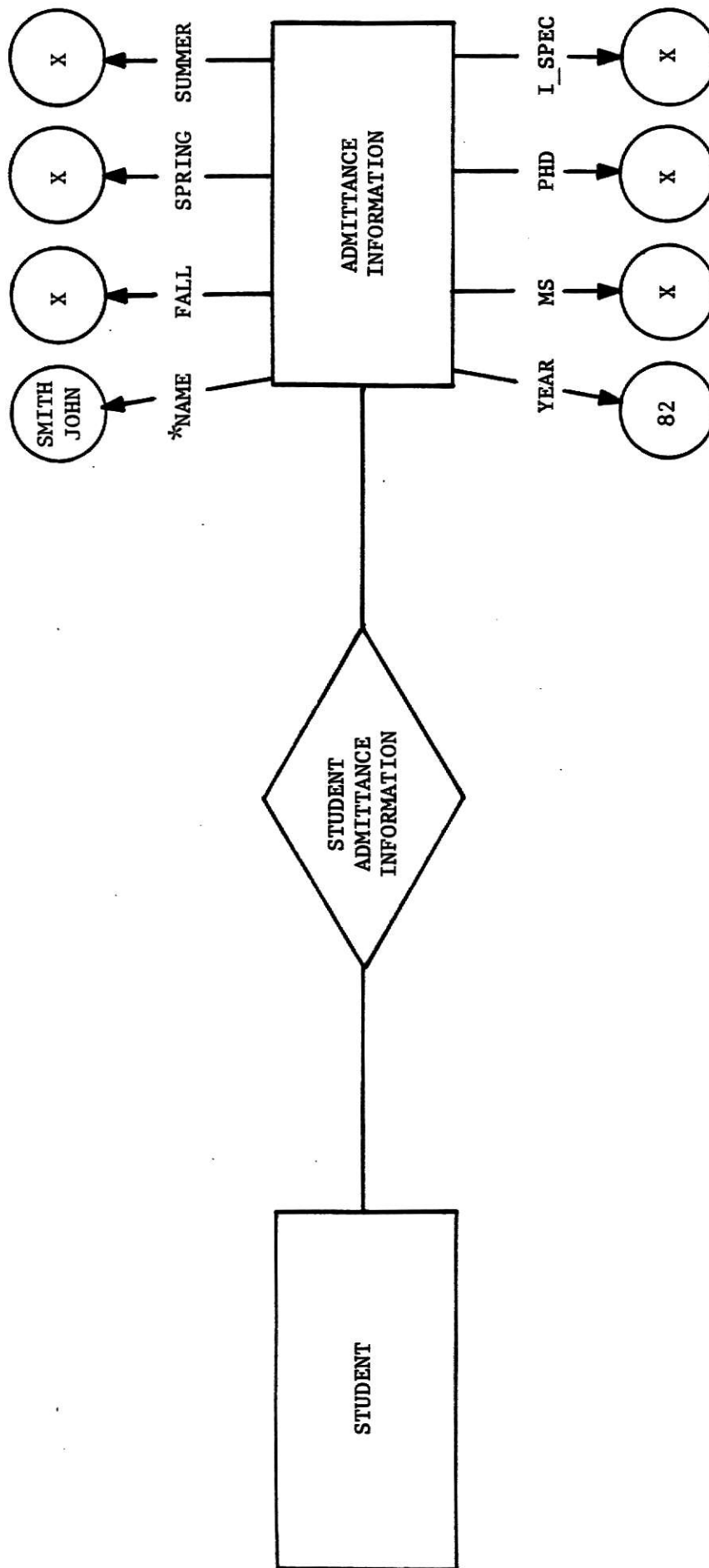


Figure B.5
STUDENT_ADMITTANCE_INFORMATION Relationship

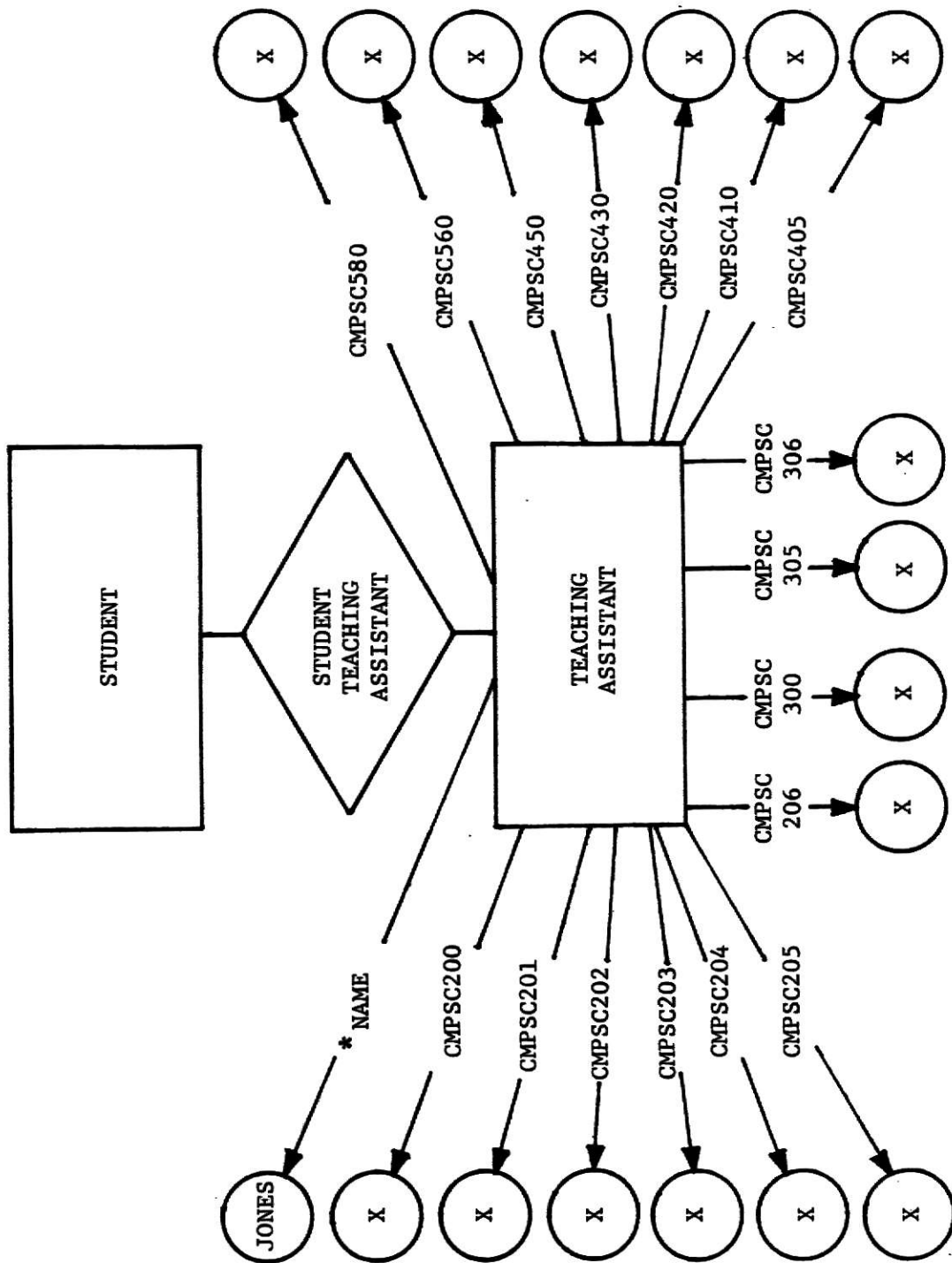


Figure B.6
STUDENT_TEACHING ASSISTANT Relationship



Figure B.7
STUDENT_MASTERS RESEARCH Relationship

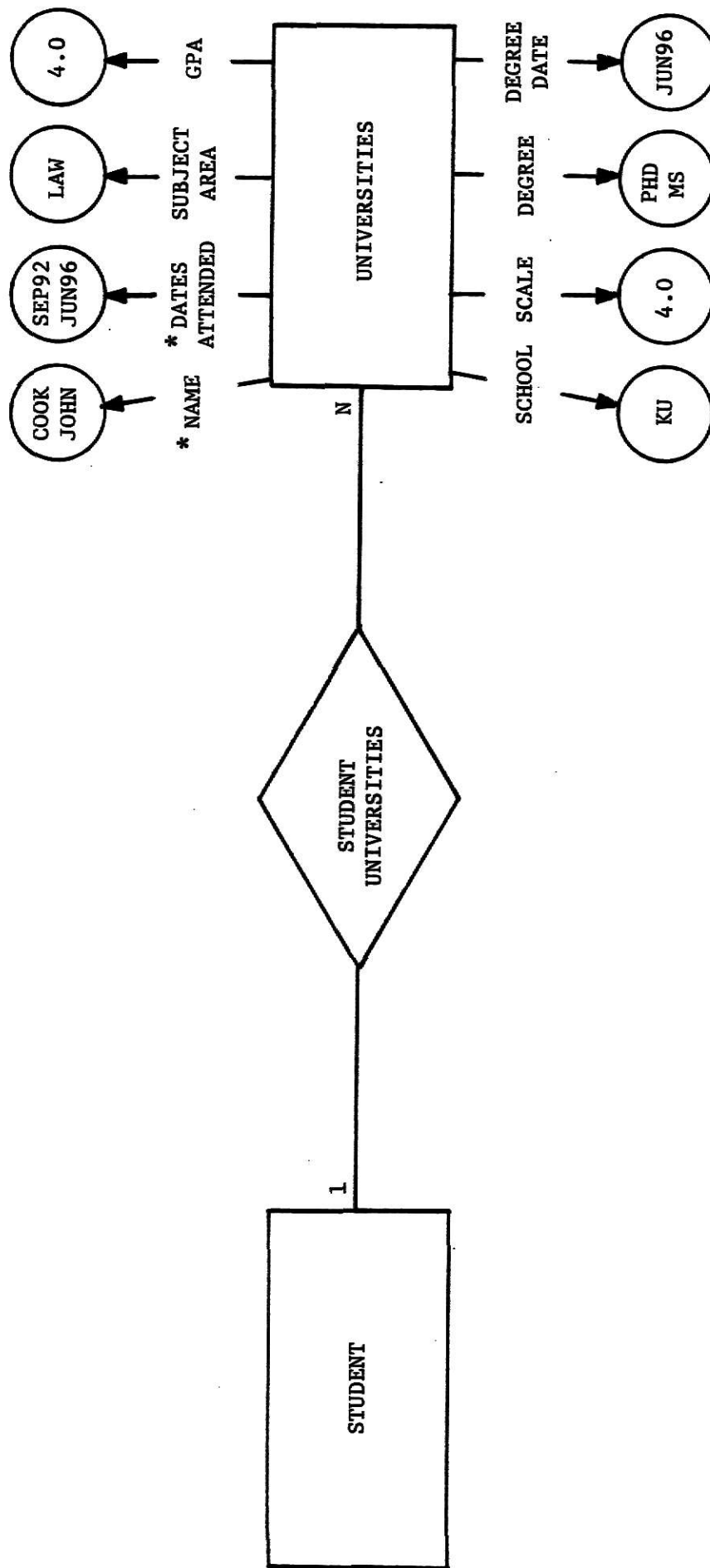


Figure B.8
STUDENT-UNIVERSITIES Relationship

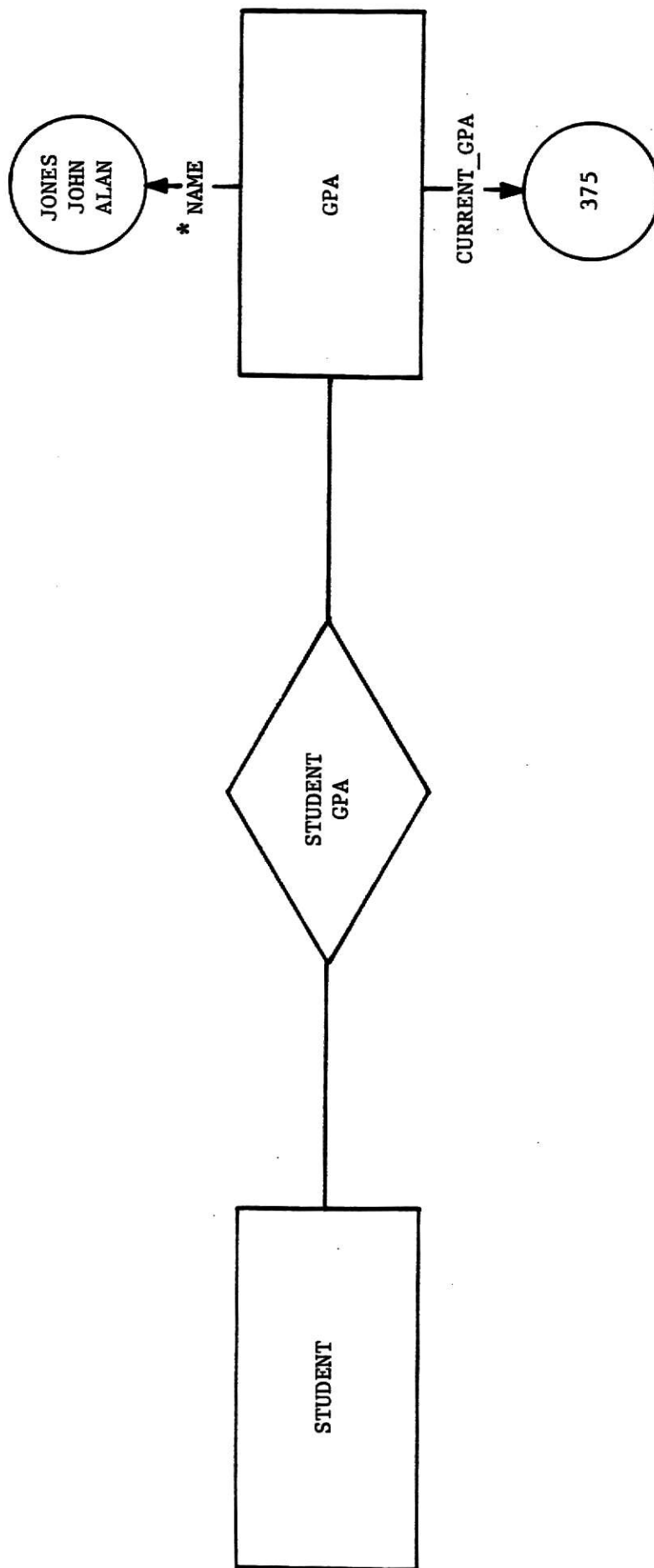


Figure B.9
STUDENT-GPA Relationship

APPENDIX C
ADBG DESIGN I

DOCUMENTS IN THE SYSTEM

1. COURSE_LIST	WITH	3 COLUMNS
2. GPA	WITH	2 COLUMNS
3. GRAD_COURSES	WITH	8 COLUMNS
4. INITIAL_ADM_INFO	WITH	8 COLUMNS
5. MASTERS_RESEARCH	WITH	14 COLUMNS
6. STUDENT_STATUS	WITH	6 COLUMNS
7. STUDENT_INFO	WITH	31 COLUMNS
8. TEACHING_ASSISTANT	WITH	18 COLUMNS
9. UNIV_ATTENDED	WITH	7 COLUMNS

DOCUMENT / COLUMN LISTING

* COURSE_LIST

DOCUMENT ATTRIBUTES :
LOC . INPCRS_NO
CRS_NAME
CREDIT

* GPA

DOCUMENT ATTRIBUTES :
LOC . INPNAME
CURRENT_GPA

* GRAD_COURSES

DOCUMENT ATTRIBUTES :
LOC . INPNAME
CRS_NO
SEM_YR
CRS_GRD
DEF
REPEAT
MAJOR
SUPPORT

* INITIAL_ADM_INFO

DOCUMENT ATTRIBUTES :
LOC . INPNAME
FALL
SPRING
SUMMER
YEAR
MS
PHD
SPEC

* MASTERS_RESEARCH

DOCUMENT ATTRIBUTES :
LOC . INPNAME
THESIS
REPORT
PAPER
PROPOSAL_DATE
PROPOSAL_APPROVED
TITLE
ORALS_DATE
PASS
FAIL
COND_PASS
MAJ_PROF

MEMBER_2
MEMBER_3

* STUDENT_STATUS

DOCUMENT ATTRIBUTES :

NAME
STATUS_DATE
REG
PROV
SPEC
PROB

LOC . INP
*
*

* STUDENT_INFO

DOCUMENT ATTRIBUTES :

NAME
SSAN
LOCAL_ADDRESS
PERMENANT_ADDRESS
LOCAL_TEL_NO
SEX
RESIDENCY_DATE
CITIZENSHIP
PREVIOUS_NAME
GRAD_APP_DATE
GRE_OFFICIAL
GRE_VERBAL_SC
GRE_QUANT_SC
GRE_ANALYTICAL_SC
ADV_GRE_OFFICIAL
ADV_GRE_FLD
ADV_GRE_SC
TOEFL_OFFICIAL
TOEFL_SC
TOEFL_DATE
FINANCE_REQ
MEDICAL_REQ
LETTERS_OF_REC
CURRICULUM
ADVISOR
PROG_ENTRY_DATE
EXP_GRAD_DATE
PROG_OFF_STUDY_APP_DATE
CS_INFO_DOC_DATE
CMPSC_APP_DOC_DATE
PROC_FEES_PD

LOC . INP
*

* TEACHING_ASSISTANT

DOCUMENT ATTRIBUTES :

NAME
CMPSC_200
CMPSC_201

LOC . INP
*

CMPSC_202
CMPSC_203
CMPSC_204
CMPSC_205
CMPSC_206
CMPSC_300
CMPSC_305
CMPSC_306
CMPSC_405
CMPSC_410
CMPSC_420
CMPSC_430
CMPSC_450
CMPSC_560
CMPSC_580

* UNIV_ATTENDED

DOCUMENT ATTRIBUTES :

LOC . INP

NAME

DATES_ATTENDED

GPA

SCALE

DEGREE

DEGREE_DATE

CURRICULUM

COLUMNS IN THE SYSTEM

1. ADV_GRE_OFFICIAL
2. ADV_GRE_FLD
3. ADV_GRE_SC
4. ADVISOR
5. CRS_NO
6. CRS_NAME
7. CREDIT
8. CURRENT_GPA
9. CRS_GRD
10. COND_PASS
11. CITIZENSHIP
12. CURRICULUM
13. CS_INFO_DOC_DATE
14. CMPSC_APP_DOC_DATE
15. CMPSC_200
16. CMPSC_201
17. CMPSC_202
18. CMPSC_203
19. CMPSC_204
20. CMPSC_205
21. CMPSC_206
22. CMPSC_300
23. CMPSC_305
24. CMPSC_306
25. CMPSC_405
26. CMPSC_410
27. CMPSC_420
28. CMPSC_430
29. CMPSC_450
30. CMPSC_560
31. CMPSC_580
32. DEF
33. DATES_ATTENDED
34. DEGREE
35. DEGREE_DATE
36. EXP_GRAD_DATE
37. FALL
38. FAIL
39. FINANCE_REQ
40. GRAD_APP_DATE
41. GRE_OFFICIAL
42. GRE_VERBAL_SC
43. GRE_QUANT_SC
44. GRE_ANALYTICAL_SC
45. GPA
46. LOCAL_ADDRESS
47. LOCAL_TEL_NO
48. LETTERS_OF_REC
49. MAJOR
50. MS
51. MAJ_PROF
52. MEMBER_2

53. MEMBER_3
54. MEDICAL_REQ
55. NAME
56. ORALS_DATE
57. PHD
58. PAPER
59. PROPOSAL_DATE
60. PROPOSAL_APPROVED
61. PASS
62. PROV
63. PROB
64. PERMENANT_ADDRESS
65. PREVIOUS_NAME
66. PROG_ENTRY_DATE
67. PROG_OFF_STUDY_APP_DATE
68. PROC_FEES_PD
69. REPEAT
70. REPORT
71. REG
72. RESIDENCY_DATE
73. SEM_YR
74. SUPPORT
75. SPRING
76. SUMMER
77. SPEC
78. STATUS_DATE
79. SSAN
80. SEX
81. SCALE
82. THESIS
83. TITLE
84. TOEFL_OFFICIAL
85. TOEFL_SC
86. TOEFL_DATE
87. YEAR

COLUMN CROSS REFERENCE

COLUMNS	IN DOCUMENTS
ADV_GRE_OFFICIAL	STUDENT_INFO
ADV_GRE_FLD	STUDENT_INFO
ADV_GRE_SC	STUDENT_INFO
ADVISOR	STUDENT_INFO
CRS_NO	COURSE_LIST GRAD_COURSES
CRS_NAME	COURSE_LIST
CREDIT	COURSE_LIST
CURRENT_GPA	GPA
CRS_GRD	GRAD_COURSES
COND_PASS	MASTERS_RESEARCH
CITIZENSHIP	STUDENT_INFO
CURRICULUM	STUDENT_INFO UNIV_ATTENDED
CS_INFO_DOC_DATE	STUDENT_INFO
CMPSC_APP_DOC_DATE	STUDENT_INFO
CMPSC_200	TEACHING_ASSISTANT
CMPSC_201	TEACHING_ASSISTANT
CMPSC_202	TEACHING_ASSISTANT
CMPSC_203	TEACHING_ASSISTANT
CMPSC_204	TEACHING_ASSISTANT
CMPSC_205	TEACHING_ASSISTANT
CMPSC_206	TEACHING_ASSISTANT
CMPSC_300	TEACHING_ASSISTANT

CMPSC_305	TEACHING_ASSISTANT
CMPSC_306	TEACHING_ASSISTANT
CMPSC_405	TEACHING_ASSISTANT
CMPSC_410	TEACHING_ASSISTANT
CMPSC_420	TEACHING_ASSISTANT
CMPSC_430	TEACHING_ASSISTANT
CMPSC_450	TEACHING_ASSISTANT
CMPSC_560	TEACHING_ASSISTANT
CMPSC_580	TEACHING_ASSISTANT
DEF	GRAD_COURSES
DATES_ATTENDED	UNIV_ATTENDED
DEGREE	UNIV_ATTENDED
DEGREE_DATE	UNIV_ATTENDED
EXP_GRAD_DATE	STUDENT_INFO
FALL	INITIAL_ADM_INFO
FAIL	MASTERS_RESEARCH
FINANCE_REQ	STUDENT_INFO
GRAD_APP_DATE	STUDENT_INFO
GRE_OFFICIAL	STUDENT_INFO
GRE_VERBAL_SC	STUDENT_INFO
GRE_QUANT_SC	STUDENT_INFO
GRE_ANALYTICAL_SC	STUDENT_INFO
GPA	UNIV_ATTENDED
LOCAL_ADDRESS	STUDENT_INFO
LOCAL_TEL_NO	STUDENT_INFO
LETTERS_OF_REC	STUDENT_INFO
MAJOR	GRAD_COURSES

MS	INITIAL_ADM_INFO
MAJ_PROF	MASTERS_RESEARCH
MEMBER_2	MASTERS_RESEARCH
MEMBER_3	MASTERS_RESEARCH
MEDICAL_REQ	STUDENT_INFO
NAME	GPA GRAD_COURSES INITIAL_ADM_INFO MASTERS_RESEARCH STUDENT_STATUS STUDENT_INFO TEACHING_ASSISTANT UNIV_ATTENDED
ORALS_DATE	MASTERS_RESEARCH
PHD	INITIAL_ADM_INFO
PAPER	MASTERS_RESEARCH
PROPOSAL_DATE	MASTERS_RESEARCH
PROPOSAL_APPROVED	MASTERS_RESEARCH
PASS	MASTERS_RESEARCH
PROV	STUDENT_STATUS
PROB	STUDENT_STATUS
PERMENANT_ADDRESS	STUDENT_INFO
PREVIOUS_NAME	STUDENT_INFO
PROG_ENTRY_DATE	STUDENT_INFO
PROG_OFF_STUDY_APP_DATE	STUDENT_INFO
PROC_FEES_PD	STUDENT_INFO
REPEAT	GRAD_COURSES
REPORT	MASTERS_RESEARCH
REG	STUDENT_STATUS
RESIDENCY_DATE	STUDENT_INFO
SEM_YR	GRAD_COURSES

SUPPORT	GRAD_COURSES
SPRING	INITIAL_ADM_INFO
SUMMER	INITIAL_ADM_INFO
SPEC	INITIAL_ADM_INFO STUDENT_STATUS
STATUS_DATE	STUDENT_STATUS
SSAN	STUDENT_INFO
SEX	STUDENT_INFO
SCALE	UNIV_ATTENDED
THESIS	MASTERS_RESEARCH
TITLE	MASTERS_RESEARCH
TOEFL_OFFICIAL	STUDENT_INFO
TOEFL_SC	STUDENT_INFO
TOEFL_DATE	STUDENT_INFO
YEAR	INITIAL_ADM_INFO

DICTIONARY - BERNSTEIN'S ALGORITHM

ADV_GRE_OFFICIAL	A00
ADV_GRE_FLD	A01
ADV_GRE_SC	A02
ADVISOR	A03
CRS_NO	C00
CRS_NAME	C01
CREDIT	C02
CURRENT_GPA	C03
CRS_GRD	C04
COND_PASS	C05
CITIZENSHIP	C06
CURRICULUM	C07
CS_INFO_DOC_DATE	C08
CMPSC_APP_DOC_DATE	C09
CMPSC_200	C10
CMPSC_201	C11
CMPSC_202	C12
CMPSC_203	C13
CMPSC_204	C14
CMPSC_205	C15
CMPSC_206	C16
CMPSC_300	C17
CMPSC_305	C18
CMPSC_306	C19
CMPSC_405	C20
CMPSC_410	C21
CMPSC_420	C22
CMPSC_430	C23
CMPSC_450	C24
CMPSC_560	C25
CMPSC_580	C26
DEF	D00
DATES_ATTENDED	D01
DEGREE	D02
DEGREE_DATE	D03
EXP_GRAD_DATE	E00
FALL	F00
FAIL	F01
FINANCE_REQ	F02
GRAD_APP_DATE	G00
GRE_OFFICIAL	G01
GRE_VERBAL_SC	G02
GRE_QUANT_SC	G03
GRE_ANALYTICAL_SC	G04
GPA	G05
LOCAL_ADDRESS	L00
LOCAL_TEL_NO	L01
LETTERS_OF_REC	L02
MAJOR	M00
MS	M01
MAJ_PROF	M02
MEMBER_2	M03

MEMBER_3	M04
MEDICAL_REQ	M05
NAME	N00
ORALS_DATE	O00
PHD	P00
PAPER	P01
PROPOSAL_DATE	P02
PROPOSAL_APPROVED	P03
PASS	P04
PROV	P05
PROB	P06
PERMENANT_ADDRESS	P07
PREVIOUS_NAME	P08
PROG_ENTRY_DATE	P09
PROG_OFF_STUDY_APP_DATE	P10
PROC_FEES_PD	P11
REPEAT	R00
REPORT	R01
REG	R02
RESIDENCY_DATE	R03
SEM_YR	S00
SUPPORT	S01
SPRING	S02
SUMMER	S03
SPEC	S04
STATUS_DATE	S05
SSAN	S06
SEX	S07
SCALE	S08
THESIS	T00
TITLE	T01
TOEFL_OFFICIAL	T02
TOEFL_SC	T03
TOEFL_DATE	T04
YEAR	Y00

THE INPUT TO THE PROGRAM IS :

C00>C01,C02;

N00>C03;

N00,C00>S00,C04,D00,R00,M00,S01;

N00>F00,S02,S03,Y00,M01,P00,S04;

N00>T00,R01,P01,P02,P03,T01,O00,P04,F01,C05,M02,M03,M04;

N00,S05>R02,P05,S04,P06;

N00>S06,L00,P07,L01,S07,R03,C06,P08,G00,G01,G02,G03,G04,A00,A01,A02,T02
,T03,T04,F02,M05,L02,C07,A03,P09,E00,P10,C08,C09,P11;

N00>C10,C11,C12,C13,C14,C15,C16,C17,C18,C19,C20,C21,C22,C23,C24,C25,C26;

N00,D01>G05,S08,D02,D03,C07;

END.

THIS IS THE LIST OF ATTRIBUTES WITH THEIR ABBREVIATIONS.

C00	C00
C01	C01
C02	C02
N00	N00
C03	C03
S00	S00
C04	C04
D00	D00
R00	R00
M00	M00
S01	S01
F00	F00
S02	S02
S03	S03
Y00	Y00
M01	M01
P00	P00
S04	S04
T00	T00
R01	R01
P01	P01
P02	P02
P03	P03
T01	T01
O00	O00
P04	P04

F01	F01
C05	C05
M02	M02
M03	M03
M04	M04
S05	S05
R02	R02
P05	P05
P06	P06
S06	S06
L00	L00
P07	P07
L01	L01
S07	S07
R03	R03
C06	C06
P08	P08
G00	G00
G01	G01
G02	G02
G03	G03
G04	G04
A00	A00
A01	A01
A02	A02
T02	T02
T03	T03
T04	T04
F02	F02
M05	M05
L02	L02
C07	C07
A03	A03
P09	P09
E00	E00
P10	P10
C08	C08
C09	C09
P11	P11
C10	C10
C11	C11
C12	C12
C13	C13
C14	C14
C15	C15
C16	C16
C17	C17
C18	C18
C19	C19
C20	C20
C21	C21
C22	C22
C23	C23
C24	C24

C25 C25
C26 C26
D01 D01
G05 G05
S08 S08
D02 D02
D03 D03

THE TOKENS MARKED *TRUE* ARE EXTRANEIOUS IN THE FDS :

FD NUMBER :001 TOKEN: C00
F
FD NUMBER :002 TOKEN: C00
F
FD NUMBER :003 TOKEN: N00
F
FD NUMBER :004 TOKEN: C00
F
FD NUMBER :004 TOKEN: N00
F
FD NUMBER :005 TOKEN: N00
F
FD NUMBER :005 TOKEN: C00
F
FD NUMBER :006 TOKEN: C00
F
FD NUMBER :006 TOKEN: N00
F
FD NUMBER :007 TOKEN: N00
F
FD NUMBER :007 TOKEN: C00
F
FD NUMBER :008 TOKEN: C00
F
FD NUMBER :008 TOKEN: N00
F
FD NUMBER :009 TOKEN: N00
F
FD NUMBER :009 TOKEN: C00
F
FD NUMBER :010 TOKEN: N00
F
FD NUMBER :011 TOKEN: N00
F
FD NUMBER :012 TOKEN: N00
F
FD NUMBER :013 TOKEN: N00
F
FD NUMBER :014 TOKEN: N00
F
FD NUMBER :015 TOKEN: N00
F
FD NUMBER :016 TOKEN: N00
F
FD NUMBER :017 TOKEN: N00

F
FD NUMBER :018 TOKEN: N00
F
FD NUMBER :019 TOKEN: N00
F
FD NUMBER :020 TOKEN: N00
F
FD NUMBER :021 TOKEN: N00
F
FD NUMBER :022 TOKEN: N00
F
FD NUMBER :023 TOKEN: N00
F
FD NUMBER :024 TOKEN: N00
F
FD NUMBER :025 TOKEN: N00
F
FD NUMBER :026 TOKEN: N00
F
FD NUMBER :027 TOKEN: N00
F
FD NUMBER :028 TOKEN: N00
F
FD NUMBER :029 TOKEN: N00
F
FD NUMBER :030 TOKEN: S05
F
FD NUMBER :030 TOKEN: N00
F
FD NUMBER :031 TOKEN: N00
F
FD NUMBER :031 TOKEN: S05
F
FD NUMBER :032 TOKEN: S05
T
FD NUMBER :032 TOKEN: N00
F
FD NUMBER :033 TOKEN: N00
F
FD NUMBER :033 TOKEN: S05
F
FD NUMBER :034 TOKEN: N00
F
FD NUMBER :035 TOKEN: N00
F
FD NUMBER :036 TOKEN: N00
F
FD NUMBER :037 TOKEN: N00
F
FD NUMBER :038 TOKEN: N00
F
FD NUMBER :039 TOKEN: N00
F
FD NUMBER :040 TOKEN: N00

F
FD NUMBER :041 TOKEN: N00
F
FD NUMBER :042 TOKEN: N00
F
FD NUMBER :043 TOKEN: N00
F
FD NUMBER :044 TOKEN: N00
F
FD NUMBER :045 TOKEN: N00
F
FD NUMBER :046 TOKEN: N00
F
FD NUMBER :047 TOKEN: N00
F
FD NUMBER :048 TOKEN: N00
F
FD NUMBER :049 TOKEN: N00
F
FD NUMBER :050 TOKEN: N00
F
FD NUMBER :051 TOKEN: N00
F
FD NUMBER :052 TOKEN: N00
F
FD NUMBER :053 TOKEN: N00
F
FD NUMBER :054 TOKEN: N00
F
FD NUMBER :055 TOKEN: N00
F
FD NUMBER :056 TOKEN: N00
F
FD NUMBER :057 TOKEN: N00
F
FD NUMBER :058 TOKEN: N00
F
FD NUMBER :059 TOKEN: N00
F
FD NUMBER :060 TOKEN: N00
F
FD NUMBER :061 TOKEN: N00
F
FD NUMBER :062 TOKEN: N00
F
FD NUMBER :063 TOKEN: N00
F
FD NUMBER :064 TOKEN: N00
F
FD NUMBER :065 TOKEN: N00
F
FD NUMBER :066 TOKEN: N00
F
FD NUMBER :067 TOKEN: N00

F
FD NUMBER :068 TOKEN: N00
F
FD NUMBER :069 TOKEN: N00
F
FD NUMBER :070 TOKEN: N00
F
FD NUMBER :071 TOKEN: N00
F
FD NUMBER :072 TOKEN: N00
F
FD NUMBER :073 TOKEN: N00
F
FD NUMBER :074 TOKEN: N00
F
FD NUMBER :075 TOKEN: N00
F
FD NUMBER :076 TOKEN: N00
F
FD NUMBER :077 TOKEN: N00
F
FD NUMBER :078 TOKEN: N00
F
FD NUMBER :079 TOKEN: N00
F
FD NUMBER :080 TOKEN: N00
F
FD NUMBER :081 TOKEN: D01
F
FD NUMBER :081 TOKEN: N00
F
FD NUMBER :082 TOKEN: N00
F
FD NUMBER :082 TOKEN: D01
F
FD NUMBER :083 TOKEN: D01
F
FD NUMBER :083 TOKEN: N00
F
FD NUMBER :084 TOKEN: N00
F
FD NUMBER :084 TOKEN: D01
F
FD NUMBER :085 TOKEN: D01
T
FD NUMBER :085 TOKEN: N00
F
THE REDUNDANT FDS ARE MARKED *TRUE* :
FD-NUMBR001
F
FD-NUMBR002
F
FD-NUMBR003
F

FD-NUMBR004
F
FD-NUMBR005
F
FD-NUMBR006
F
FD-NUMBR007
F
FD-NUMBR008
F
FD-NUMBR009
F
FD-NUMBR010
F
FD-NUMBR011
F
FD-NUMBR012
F
FD-NUMBR013
F
FD-NUMBR014
F
FD-NUMBR015
F
FD-NUMBR016
T
FD-NUMBR017
F
FD-NUMBR018
F
FD-NUMBR019
F
FD-NUMBR020
F
FD-NUMBR021
F
FD-NUMBR022
F
FD-NUMBR023
F
FD-NUMBR024
F
FD-NUMBR025
F
FD-NUMBR026
F
FD-NUMBR027
F
FD-NUMBR028
F
FD-NUMBR029
F
FD-NUMBR030
F

FD-NUMBR031
F
FD-NUMBR032
F
FD-NUMBR033
F
FD-NUMBR034
F
FD-NUMBR035
F
FD-NUMBR036
F
FD-NUMBR037
F
FD-NUMBR038
F
FD-NUMBR039
F
FD-NUMBR040
F
FD-NUMBR041
F
FD-NUMBR042
F
FD-NUMBR043
F
FD-NUMBR044
F
FD-NUMBR045
F
FD-NUMBR046
F
FD-NUMBR047
F
FD-NUMBR048
F
FD-NUMBR049
F
FD-NUMBR050
F
FD-NUMBR051
F
FD-NUMBR052
F
FD-NUMBR053
F
FD-NUMBR054
F
FD-NUMBR055
F
FD-NUMBR056
T
FD-NUMBR057
F

FD-NUMBR058
F
FD-NUMBR059
F
FD-NUMBR060
F
FD-NUMBR061
F
FD-NUMBR062
F
FD-NUMBR063
F
FD-NUMBR064
F
FD-NUMBR065
F
FD-NUMBR066
F
FD-NUMBR067
F
FD-NUMBR068
F
FD-NUMBR069
F
FD-NUMBR070
F
FD-NUMBR071
F
FD-NUMBR072
F
FD-NUMBR073
F
FD-NUMBR074
F
FD-NUMBR075
F
FD-NUMBR076
F
FD-NUMBR077
F
FD-NUMBR078
F
FD-NUMBR079
F
FD-NUMBR080
F
FD-NUMBR081
F
FD-NUMBR082
F
FD-NUMBR083
F
FD-NUMBR084
F

FD-NUMBER085

F

THE FOLLOWING FDS HAVE THE SAME LHS AND ARE THEREFORE GROUPED TOGETHER INTO PARTITION CLASSES:

PARTITION CLASS-NUMBER 001:084083082081

PARTITION CLASS-NUMBER 002:033031030

PARTITION CLASS-NUMBER 003:009008007006005004

PARTITION CLASS-NUMBER 004:00301001101201301401501701801902002102202302

40250260270280290320340350360370380390400410420430440450460704804905005

105205305405505705805906006106206306406506606706806907007107

PARTITION CLASS-NUMBER 005:001002

004

001

002

005

003

THE FOLLOWING FDS ARE REDUNDANT AFTER ADDING THE BIJECTIONS TO THE FD STRUCTURE :

THIS IS THE SCHEMA IN 3NF :

(N00) >	C03	F00	S02	S03	Y00	M01	P00	T00	R01	P01	P02	P03
	T01	000	P04	F01	C05	M02	M03	M04	S04	S06	L00	P07
	L01	S07	R03	C06	P08	G00	G01	G02	G03	G04	A00	A01
	A02	T02	T03	T04	F02	M05	L02	A03	P09	E00	P10	C08
	C09	P11	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19
	C20	C21	C22	C23	C24	C25	C26	C07				

(N00D01) > D03 D02 S08 G05

(N00S05) > P06 P05 R02

(C00) > C01 C02

(N00C00) > S01 M00 R00 D00 C04 S00

APPENDIX D
ADBG DESIGN II

DOCUMENTS IN THE SYSTEM

1. COURSE_LIST	WITH	3 COLUMNS
2. GPA	WITH	2 COLUMNS
3. GRAD_COURSES	WITH	8 COLUMNS
4. INITIAL_ADM_INFO	WITH	8 COLUMNS
5. MASTERS_RESEARCH	WITH	14 COLUMNS
6. STUDENT_STATUS	WITH	6 COLUMNS
7. STUDENT_INFO	WITH	31 COLUMNS
8. TEACHING_ASSISTANT	WITH	18 COLUMNS
9. UNIV_ATTENDED	WITH	8 COLUMNS

DOCUMENT / COLUMN LISTING

* COURSE_LIST

DOCUMENT ATTRIBUTES :

LOC . INP

*

CRS_NO
CRS_NAME
CREDIT

* GPA

DOCUMENT ATTRIBUTES :

LOC . INP

*

NAME
CURRENT_GPA

* GRAD_COURSES

DOCUMENT ATTRIBUTES :

LOC . INP

*

NAME
CRS_NO
SEM_YR
CRS_GRD
DEF
REPEAT
MAJOR
SUPPORT

* INITIAL_ADM_INFO

DOCUMENT ATTRIBUTES :

LOC . INP

*

NAME
FALL
SPRING
SUMMER
YEAR
MS
PHD
I_SPEC

* MASTERS_RESEARCH

DOCUMENT ATTRIBUTES :

LOC . INP

*

NAME
THESIS
REPORT
PAPER
PROPOSAL_DATE
PROPOSAL_APPROVED
TITLE
ORALS_DATE
PASS
FAIL
COND_PASS
MAJ_PROF
MEMBER_2

MEMBER_3

* STUDENT_STATUS

DOCUMENT ATTRIBUTES :

LOC . INP

NAME *

STATUS_DATE *

REG

PROV

SPEC

PROB

* STUDENT_INFO

DOCUMENT ATTRIBUTES :

LOC . INP

NAME *

SSAN

LOCAL_ADDRESS

PERMENANT_ADDRESS

LOCAL_TEL_NO

SEX

RESIDENCY_DATE

CITIZENSHIP

PREVIOUS_NAME

GRAD_APP_DATE

GRE_OFFICIAL

GRE_VERBAL_SC

GRE_QUANT_SC

GRE_ANALYTICAL_SC

ADV_GRE_OFFICIAL

ADV_GRE_FLD

ADV_GRE_SC

TOEFL_OFFICIAL

TOEFL_SC

TOEFL_DATE

FINANCE_REQ

MEDICAL_REQ

LETTERS_OF_REC

CURRICULUM

ADVISOR

PROG_ENTRY_DATE

EXP_GRAD_DATE

PROG_OFF_STUDY_APP_DATE

CS_INFO_DOC_DATE

CMPSC_APP_DOC_DATE

PROC_FEES_PD

* TEACHING_ASSISTANT

DOCUMENT ATTRIBUTES :

LOC . INP

NAME *

CMPSC_200

CMPSC_201

CMPSC_202

CMPSC_203

CMPSC_204
CMPSC_205
CMPSC_206
CMPSC_300
CMPSC_305
CMPSC_306
CMPSC_405
CMPSC_410
CMPSC_420
CMPSC_430
CMPSC_450
CMPSC_560
CMPSC_580

* UNIV_ATTENDED

DOCUMENT ATTRIBUTES :

LOC . INP

NAME

DATES_ATTENDED

SCHOOL

SUBJECT_AREA

GPA

SCALE

DEGREE

DEGREE_DATE

COLUMNS IN THE SYSTEM

1. ADV_GRE_OFFICIAL
2. ADV_GRE_FLD
3. ADV_GRE_SC
4. ADVISOR
5. CRS_NO
6. CRS_NAME
7. CREDIT
8. CURRENT_GPA
9. CRS_GRD
10. COND_PASS
11. CITIZENSHIP
12. CURRICULUM
13. CS_INFO_DOC_DATE
14. CMPSC_APP_DOC_DATE
15. CMPSC_200
16. CMPSC_201
17. CMPSC_202
18. CMPSC_203
19. CMPSC_204
20. CMPSC_205
21. CMPSC_206
22. CMPSC_300
23. CMPSC_305
24. CMPSC_306
25. CMPSC_405
26. CMPSC_410
27. CMPSC_420
28. CMPSC_430
29. CMPSC_450
30. CMPSC_560
31. CMPSC_580
32. DEF
33. DATES_ATTENDED
34. DEGREE
35. DEGREE_DATE
36. EXP_GRAD_DATE
37. FALL
38. FAIL
39. FINANCE_REQ
40. GRAD_APP_DATE
41. GRE_OFFICIAL
42. GRE_VERBAL_SC
43. GRE_QUANT_SC
44. GRE_ANALYTICAL_SC
45. GPA
46. I_SPEC
47. LOCAL_ADDRESS
48. LOCAL_TEL_NO
49. LETTERS_OF_REC
50. MAJOR
51. MS
52. MAJ_PROF
53. MEMBER_2

54. MEMBER_3
55. MEDICAL_REQ
56. NAME
57. ORALS_DATE
58. PHD
59. PAPER
60. PROPOSAL_DATE
61. PROPOSAL_APPROVED
62. PASS
63. PROV
64. PROB
65. PERMENANT_ADDRESS
66. PREVIOUS_NAME
67. PROG_ENTRY_DATE
68. PROG_OFF_STUDY_APP_DATE
69. PROC_FEES_PD
70. REPEAT
71. REPORT
72. REG
73. RESIDENCY_DATE
74. SEM_YR
75. SUPPORT
76. SPRING
77. SUMMER
78. STATUS_DATE
79. SPEC
80. SSAN
81. SEX
82. SCHOOL
83. SUBJECT_AREA
84. SCALE
85. THESIS
86. TITLE
87. TOEFL_OFFICIAL
88. TOEFL_SC
89. TOEFL_DATE
90. YEAR

COLUMN CROSS REFERENCE

COLUMNS	IN DOCUMENTS
ADV_GRE_OFFICIAL	STUDENT_INFO
ADV_GRE_FLD	STUDENT_INFO
ADV_GRE_SC	STUDENT_INFO
ADVISOR	STUDENT_INFO
CRS_NO	COURSE_LIST GRAD_COURSES
CRS_NAME	COURSE_LIST
CREDIT	COURSE_LIST
CURRENT_GPA	GPA
CRS_GRD	GRAD_COURSES
COND_PASS	MASTERS_RESEARCH
CITIZENSHIP	STUDENT_INFO
CURRICULUM	STUDENT_INFO
CS_INFO_DOC_DATE	STUDENT_INFO
CMPSC_APP_DOC_DATE	STUDENT_INFO
CMPSC_200	TEACHING_ASSISTANT
CMPSC_201	TEACHING_ASSISTANT
CMPSC_202	TEACHING_ASSISTANT
CMPSC_203	TEACHING_ASSISTANT
CMPSC_204	TEACHING_ASSISTANT
CMPSC_205	TEACHING_ASSISTANT
CMPSC_206	TEACHING_ASSISTANT
CMPSC_300	TEACHING_ASSISTANT
CMPSC_305	TEACHING_ASSISTANT
CMPSC_306	TEACHING_ASSISTANT

CMPSC_405	TEACHING_ASSISTANT
CMPSC_410	TEACHING_ASSISTANT
CMPSC_420	TEACHING_ASSISTANT
CMPSC_430	TEACHING_ASSISTANT
CMPSC_450	TEACHING_ASSISTANT
CMPSC_560	TEACHING_ASSISTANT
CMPSC_580	TEACHING_ASSISTANT
DEF	GRAD_COURSES
DATES_ATTENDED	UNIV_ATTENDED
DEGREE	UNIV_ATTENDED
DEGREE_DATE	UNIV_ATTENDED
EXP_GRAD_DATE	STUDENT_INFO
FALL	INITIAL_ADM_INFO
FAIL	MASTERS_RESEARCH
FINANCE_REQ	STUDENT_INFO
GRAD_APP_DATE	STUDENT_INFO
GRE_OFFICIAL	STUDENT_INFO
GRE_VERBAL_SC`	STUDENT_INFO
GRE_QUANT_SC	STUDENT_INFO
GRE_ANALYTICAL_SC	STUDENT_INFO
GPA	UNIV_ATTENDED
I_SPEC	INITIAL_ADM_INFO
LOCAL_ADDRESS	STUDENT_INFO
LOCAL_TEL_NO	STUDENT_INFO
LETTERS_OF_REC	STUDENT_INFO
MAJOR	GRAD_COURSES
MS	INITIAL_ADM_INFO
MAJ_PROF	MASTERS_RESEARCH

MEMBER_2	MASTERS_RESEARCH
MEMBER_3	MASTERS_RESEARCH
MEDICAL_REQ	STUDENT_INFO
NAME	GPA
	GRAD_COURSES
	INITIAL_ADM_INFO
	MASTERS_RESEARCH
	STUDENT_STATUS
	STUDENT_INFO
	TEACHING_ASSISTANT
	UNIV_ATTENDED
ORALS_DATE	MASTERS_RESEARCH
PHD	INITIAL_ADM_INFO
PAPER	MASTERS_RESEARCH
PROPOSAL_DATE	MASTERS_RESEARCH
PROPOSAL_APPROVED	MASTERS_RESEARCH
PASS	MASTERS_RESEARCH
PROV	STUDENT_STATUS
PROB	STUDENT_STATUS
PERMENANT_ADDRESS	STUDENT_INFO
PREVIOUS_NAME	STUDENT_INFO
PROG_ENTRY_DATE	STUDENT_INFO
PROG_OFF_STUDY_APP_DATE	STUDENT_INFO
PROC_FEES_PD	STUDENT_INFO
REPEAT	GRAD_COURSES
REPORT	MASTERS_RESEARCH
REG	STUDENT_STATUS
RESIDENCY_DATE	STUDENT_INFO
SEM_YR	GRAD_COURSES
SUPPORT	GRAD_COURSES
SPRING	INITIAL_ADM_INFO

SUMMER	INITIAL_ADM_INFO
STATUS_DATE	STUDENT_STATUS
SPEC	STUDENT_STATUS
SSAN	STUDENT_INFO
SEX	STUDENT_INFO
SCHOOL	UNIV_ATTENDED
SUBJECT_AREA	UNIV_ATTENDED
SCALE	UNIV_ATTENDED
THESIS	MASTERS_RESEARCH
TITLE	MASTERS_RESEARCH
TOEFL_OFFICIAL	STUDENT_INFO
TOEFL_SC	STUDENT_INFO
TOEFL_DATE	STUDENT_INFO
YEAR	INITIAL_ADM_INFO

DICTIONARY - BERNSTEIN'S ALGORITHM

ADV_GRE_OFFICIAL	A00
ADV_GRE_FLD	A01
ADV_GRE_SC	A02
ADVISOR	A03
CRS_NO	C00
CRS_NAME	C01
CREDIT	C02
CURRENT_GPA	C03
CRS_GRD	C04
COND_PASS	C05
CITIZENSHIP	C06
CURRICULUM	C07
CS_INFO_DOC_DATE	C08
CMPSC_APP_DOC_DATE	C09
CMPSC_200	C10
CMPSC_201	C11
CMPSC_202	C12
CMPSC_203	C13
CMPSC_204	C14
CMPSC_205	C15
CMPSC_206	C16
CMPSC_300	C17
CMPSC_305	C18
CMPSC_306	C19
CMPSC_405	C20
CMPSC_410	C21
CMPSC_420	C22
CMPSC_430	C23
CMPSC_450	C24
CMPSC_560	C25
CMPSC_580	C26
DEF	D00
DATES_ATTENDED	D01
DEGREE	D02
DEGREE_DATE	D03
EXP_GRAD_DATE	E00
FALL	F00
FAIL	F01
FINANCE_REQ	F02
GRAD_APP_DATE	G00
GRE_OFFICIAL	G01
GRE_VERBAL_SC	G02
GRE_QUANT_SC	G03
GRE_ANALYTICAL_SC	G04
GPA	G05
I_SPEC	I00
LOCAL_ADDRESS	L00
LOCAL_TEL_NO	L01
LETTERS_OF_REC	L02
MAJOR	M00
MS	M01
MAJ_PROF	M02
MEMBER_2	M03

MEMBER_3	M04
MEDICAL_REQ	M05
NAME	N00
ORALS_DATE	O00
PHD	P00
PAPER	P01
PROPOSAL_DATE	P02
PROPOSAL_APPROVED	P03
PASS	P04
PROV	P05
PROB	P06
PERMENANT_ADDRESS	P07
PREVIOUS_NAME	P08
PROG_ENTRY_DATE	P09
PROG_OFF_STUDY_APP_DATE	P10
PROC_FEES_PD	P11
REPEAT	R00
REPORT	R01
REG	R02
RESIDENCY_DATE	R03
SEM_YR	S00
SUPPORT	S01
SPRING	S02
SUMMER	S03
STATUS_DATE	S04
SPEC	S05
SSAN	S06
SEX	S07
SCHOOL	S08
SUBJECT_AREA	S09
SCALE	S10
THESIS	T00
TITLE	T01
TOEFL_OFFICIAL	T02
TOEFL_SC	T03
TOEFL_DATE	T04
YEAR	Y00

THE INPUT TO THE PROGRAM IS :

C00>C01,C02;

N00>C03;

N00,C00>S00,C04,D00,R00,M00,S01;

N00>F00,S02,S03,Y00,M01,P00,I00;

N00>T00,R01,P01,P02,P03,T01,O00,P04,F01,C05,M02,M03,M04;

N00,S04>R02,P05,S05,P06;

N00>S06,L00,P07,L01,S07,R03,C06,P08,G00,G01,G02,G03,G04,A00,A01,A02,T02
,T03,T04,F02,M05,L02,C07,A03,P09,E00,P10,C08,C09,P11;

N00>C10,C11,C12,C13,C14,C15,C16,C17,C18,C19,C20,C21,C22,C23,C24,C25,C26;

N00,D01>S08,S09,G05,S10,D02,D03;

END.

THIS IS THE LIST OF ATTRIBUTES WITH THEIR ABBREVIATIONS.

C00	C00
C01	C01
C02	C02
N00	N00
C03	C03
S00	S00
C04	C04
D00	D00
R00	R00
M00	M00
S01	S01
F00	F00
S02	S02
S03	S03
Y00	Y00
M01	M01
P00	P00
I00	I00
T00	T00
R01	R01
P01	P01
P02	P02
P03	P03
T01	T01
O00	O00
P04	P04
F01	F01

C05	C05
M02	M02
M03	M03
M04	M04
S04	S04
R02	R02
P05	P05
S05	S05
P06	P06
S06	S06
L00	L00
P07	P07
L01	L01
S07	S07
R03	R03
C06	C06
P08	P08
G00	G00
G01	G01
G02	G02
G03	G03
G04	G04
A00	A00
A01	A01
A02	A02
T02	T02
T03	T03
T04	T04
F02	F02
M05	M05
L02	L02
C07	C07
A03	A03
P09	P09
E00	E00
P10	P10
C08	C08
C09	C09
P11	P11
C10	C10
C11	C11
C12	C12
C13	C13
C14	C14
C15	C15
C16	C16
C17	C17
C18	C18
C19	C19
C20	C20
C21	C21
C22	C22
C23	C23
C24	C24
C25	C25

C26 C26
D01 D01
S08 S08
S09 S09
G05 G05
S10 S10
D02 D02
D03 D03

THE TOKENS MARKED *TRUE* ARE EXTRANEIOUS IN THE FDS :

FD NUMBER :001 TOKEN: C00
F
FD NUMBER :002 TOKEN: C00
F
FD NUMBER :003 TOKEN: N00
F
FD NUMBER :004 TOKEN: C00
F
FD NUMBER :004 TOKEN: N00
F
FD NUMBER :005 TOKEN: N00
F
FD NUMBER :005 TOKEN: C00
F
FD NUMBER :006 TOKEN: C00
F
FD NUMBER :006 TOKEN: N00
F
FD NUMBER :007 TOKEN: N00
F
FD NUMBER :007 TOKEN: C00
F
FD NUMBER :008 TOKEN: C00
F
FD NUMBER :008 TOKEN: N00
F
FD NUMBER :009 TOKEN: N00
F
FD NUMBER :009 TOKEN: C00
F
FD NUMBER :010 TOKEN: N00
F
FD NUMBER :011 TOKEN: N00
F
FD NUMBER :012 TOKEN: N00
F
FD NUMBER :013 TOKEN: N00
F
FD NUMBER :014 TOKEN: N00
F
FD NUMBER :015 TOKEN: N00
F
FD NUMBER :016 TOKEN: N00
F
FD NUMBER :017 TOKEN: N00

F
FD NUMBER :018 TOKEN: N00
F
FD NUMBER :019 TOKEN: N00
F
FD NUMBER :020 TOKEN: N00
F
FD NUMBER :021 TOKEN: N00
F
FD NUMBER :022 TOKEN: N00
F
FD NUMBER :023 TOKEN: N00
F
FD NUMBER :024 TOKEN: N00
F
FD NUMBER :025 TOKEN: N00
F
FD NUMBER :026 TOKEN: N00
F
FD NUMBER :027 TOKEN: N00
F
FD NUMBER :028 TOKEN: N00
F
FD NUMBER :029 TOKEN: N00
F
FD NUMBER :030 TOKEN: S04
F
FD NUMBER :030 TOKEN: N00
F
FD NUMBER :031 TOKEN: N00
F
FD NUMBER :031 TOKEN: S04
F
FD NUMBER :032 TOKEN: S04
F
FD NUMBER :032 TOKEN: N00
F
FD NUMBER :033 TOKEN: N00
F
FD NUMBER :033 TOKEN: S04
F
FD NUMBER :034 TOKEN: N00
F
FD NUMBER :035 TOKEN: N00
F
FD NUMBER :036 TOKEN: N00
F
FD NUMBER :037 TOKEN: N00
F
FD NUMBER :038 TOKEN: N00
F
FD NUMBER :039 TOKEN: N00
F
FD NUMBER :040 TOKEN: N00
F

FD NUMBER :041 TOKEN: N00
F
FD NUMBER :042 TOKEN: N00
F
FD NUMBER :043 TOKEN: N00
F
FD NUMBER :044 TOKEN: N00
F
FD NUMBER :045 TOKEN: N00
F
FD NUMBER :046 TOKEN: N00
F
FD NUMBER :047 TOKEN: N00
F
FD NUMBER :048 TOKEN: N00
F
FD NUMBER :049 TOKEN: N00
F
FD NUMBER :050 TOKEN: N00
F
FD NUMBER :051 TOKEN: N00
F
FD NUMBER :052 TOKEN: N00
F
FD NUMBER :053 TOKEN: N00
F
FD NUMBER :054 TOKEN: N00
F
FD NUMBER :055 TOKEN: N00
F
FD NUMBER :056 TOKEN: N00
F
FD NUMBER :057 TOKEN: N00
F
FD NUMBER :058 TOKEN: N00
F
FD NUMBER :059 TOKEN: N00
F
FD NUMBER :060 TOKEN: N00
F
FD NUMBER :061 TOKEN: N00
F
FD NUMBER :062 TOKEN: N00
F
FD NUMBER :063 TOKEN: N00
F
FD NUMBER :064 TOKEN: N00
F
FD NUMBER :065 TOKEN: N00
F
FD NUMBER :066 TOKEN: N00
F
FD NUMBER :067 TOKEN: N00
F
FD NUMBER :068 TOKEN: N00

F
FD NUMBER :069 TOKEN: N00
F
FD NUMBER :070 TOKEN: N00
F
FD NUMBER :071 TOKEN: N00
F
FD NUMBER :072 TOKEN: N00
F
FD NUMBER :073 TOKEN: N00
F
FD NUMBER :074 TOKEN: N00
F
FD NUMBER :075 TOKEN: N00
F
FD NUMBER :076 TOKEN: N00
F
FD NUMBER :077 TOKEN: N00
F
FD NUMBER :078 TOKEN: N00
F
FD NUMBER :079 TOKEN: N00
F
FD NUMBER :080 TOKEN: N00
F
FD NUMBER :081 TOKEN: D01
F
FD NUMBER :081 TOKEN: N00
F
FD NUMBER :082 TOKEN: N00
F
FD NUMBER :082 TOKEN: D01
F
FD NUMBER :083 TOKEN: D01
F
FD NUMBER :083 TOKEN: N00
F
FD NUMBER :084 TOKEN: N00
F
FD NUMBER :084 TOKEN: D01
F
FD NUMBER :085 TOKEN: D01
F
FD NUMBER :085 TOKEN: N00
F
FD NUMBER :086 TOKEN: N00
F
FD NUMBER :086 TOKEN: D01
F
THE REDUNDANT FDS ARE MARKED *TRUE* :
FD-NUMBR001
F
FD-NUMBR002
F
FD-NUMBR003

F
FD-NUMBR004
F
FD-NUMBR005
F
FD-NUMBR006
F
FD-NUMBR007
F
FD-NUMBR008
F
FD-NUMBR009
F
FD-NUMBR010
F
FD-NUMBR011
F
FD-NUMBR012
F
FD-NUMBR013
F
FD-NUMBR014
F
FD-NUMBR015
F
FD-NUMBR016
F
FD-NUMBR017
F
FD-NUMBR018
F
FD-NUMBR019
F
FD-NUMBR020
F
FD-NUMBR021
F
FD-NUMBR022
F
FD-NUMBR023
F
FD-NUMBR024
F
FD-NUMBR025
F
FD-NUMBR026
F
FD-NUMBR027
F
FD-NUMBR028
F
FD-NUMBR029
F
FD-NUMBR030
F

FD-NUMBR031
F
FD-NUMBR032
F
FD-NUMBR033
F
FD-NUMBR034
F
FD-NUMBR035
F
FD-NUMBR036
F
FD-NUMBR037
F
FD-NUMBR038
F
FD-NUMBR039
F
FD-NUMBR040
F
FD-NUMBR041
F
FD-NUMBR042
F
FD-NUMBR043
F
FD-NUMBR044
F
FD-NUMBR045
F
FD-NUMBR046
F
FD-NUMBR047
F
FD-NUMBR048
F
FD-NUMBR049
F
FD-NUMBR050
F
FD-NUMBR051
F
FD-NUMBR052
F
FD-NUMBR053
F
FD-NUMBR054
F
FD-NUMBR055
F
FD-NUMBR056
F
FD-NUMBR057
F
FD-NUMBR058

F
FD-NUMBR059
F
FD-NUMBR060
F
FD-NUMBR061
F
FD-NUMBR062
F
FD-NUMBR063
F
FD-NUMBR064
F
FD-NUMBR065
F
FD-NUMBR066
F
FD-NUMBR067
F
FD-NUMBR068
F
FD-NUMBR069
F
FD-NUMBR070
F
FD-NUMBR071
F
FD-NUMBR072
F
FD-NUMBR073
F
FD-NUMBR074
F
FD-NUMBR075
F
FD-NUMBR076
F
FD-NUMBR077
F
FD-NUMBR078
F
FD-NUMBR079
F
FD-NUMBR080
F
FD-NUMBR081
F
FD-NUMBR082
F
FD-NUMBR083
F
FD-NUMBR084
F
FD-NUMBR085
F

APPENDIX E
FINAL E-R DIAGRAM

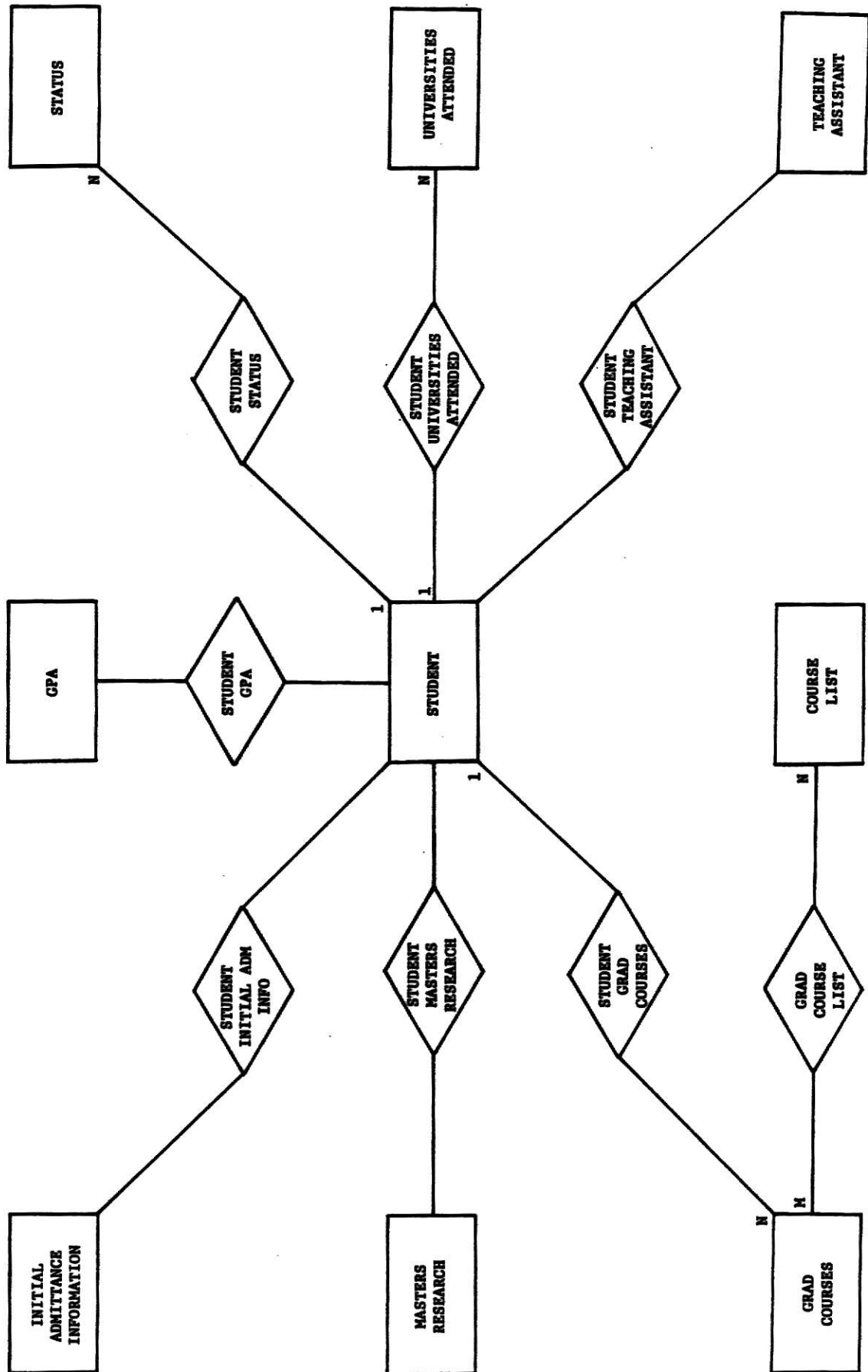


Figure E.1
Final E-R Diagram

GRADUATE STUDENT RECORDS RELATIONAL
DATA BASE DESIGN

by

JOHN L. COOK

B.S., Embry Riddle Aeronautical University, 1978

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1982

ABSTRACT

This report consists of the design of a relational data base for the Computer Science Department. The data base was designed to contain all the data required to perform administrative and counseling tasks for perspective and enrolled students.

The logical data base design was achieved using the Entity-Relationship (E-R) approach. The E-R approach to logical data base design consists of two major phases: (1) defining the enterprise schema using the Entity-Relationship diagram, and (2) translating the enterprise schema into a user schema. The enterprise schema provides a medium for understanding, and facilitates communication between the designer, data base administrator, and other management personnel.

The logical data base design was translated to a schema of normalized relations with the aid of the Automatic Data Base Generator (ADBG) system. This system aids the system's analyst in constructing a data base by permitting him to perform certain manipulations and tests on the proposed data elements. The ADBG system implements Bernstein's algorithm #2. The system transforms a set of normalized or unnormalized relations into a set of relations that are minimum in number and in third normal form (3NF).

The report presents a thorough analysis of the relations that comprise the schema designed for the Graduate Student Records Data Base and concludes with a review of the design methodology and suggestions for future enhancements of the system.