

27
A STUDY OF THE USE OF DOMAIN KEY
NORMAL FORM CRITERIA IN DATABASE DESIGN

by

RICHARD BRYAN THOMPSON

B.S., University of North Dakota, 1970

A REPORT

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

COMPUTER SCIENCE

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1988

Approved by:



LD
26608
.R4
CMSC
1988
T56
C.2

Table of Contents

A11208 232173

Chapter		Page
1	Introduction	1-1
	Goals	1-2
	Problems Solved	1-2
	Organization of the Report	1-3
2	Relevant Research	2-1
	Introduction	2-1
	Design	2-2
	Normal Forms	2-3
	KSU Adhoc Design Method	2-6
	Domain Key Example	2-12
	Conclusion	2-14
3	Application Environment	3-1
	Introduction	3-1
	Characteristics of the Environment	3-1
	Potential for Database Application within the Department	3-3
	Cadet Management	3-4
	Recruitment	3-7
	Conclusion	3-11
4	Application System Design and Implementation	4-1
	Introduction	4-1
	The Manual System	4-1
	The Analysis	4-3
	The DBMS	4-6
	The Implementation	4-10
	The Programs	4-13
	Conclusion	4-22
5	Conclusions and Future Work	5-1
	Conclusions	5-1
	Future Work	5-1
	Theoretical	5-2
	Implementation	5-2

Bibliography

Appendix A	PROSPECT Structure
Appendix B	FOLLOWUP Structure
Appendix C	CONTACT Structure
Appendix D	Data Dictionary

Appendix E	State Codes
Appendix F	Cadre Representative Report
Appendix G	Last Contact Report
Appendix H	High School Graduation Year Report
Appendix I	Region Report
Appendix J	High School Report
Appendix K	City Report
Appendix L	Scholarship Type Report
Appendix M	Source Report
Appendix N	Alphabetical Report
Appendix O	Label Report
Appendix P	Individual Report
Appendix Q	Contract Representative Report
Appendix R	START.PRG Code
Appendix S	EDITER.PRG Code
Appendix T	REPORTS.PRG Code
Appendix U	RPTFM1.PRG Code
Appendix V	LABELIT.PRG Code
Appendix W	FOLLOWUP.PRG Code
Appendix X	CONTACTS.PRG Code
Appendix Y	INDIV.PRG Code
Appendix Z	LCON.PRG Code
Appendix AA	NFOL.PRG Code
Appendix BB	CONREP1.PRG Code
Appendix CC	User's Guide and Documentation
Appendix DD	Definitions

List of Figures

Figure	Title	Page
2.1	Dependencies	2-8
2.2	Traditional Decomposition	2-9
2.3	KSU Adhoc Design Method	2-9
2.4	The Schema in 3NF (FDs only)	2-10
2.5	Relations from the KSU Adhoc Design Method	2-11
2.6	Domain Definitions	2-13
4.1	Data Sheet	4-4
4.2	ROTC Report Requirements	4-7
4.3	Dependencies	4-8
4.4	Initial Schemas	4-9
4.5	Program Hierarchy	4-12

Chapter 1

Introduction

This study focuses on the problem of database design. It attempts to provide a tool for designers which will facilitate the design of a database at the highest possible normal form.

There is a great deal of theory which surrounds the design process. Much of this theory centers on the different levels of normal forms which are achieved through the design process and the advantages and disadvantages of each level of normal form. Since Domain Key Normal Form is at the highest level, it is considered the goal of many designs. This study suggests a method of achieving Domain Key Normal Form by using a design method known as the KSU Adhoc Design Method. The study is supported by an actual design and implementation of an automated database for the KSU Army ROTC Department.

This chapter addresses the goals of this study, and it briefly discusses the problems solved. The chapter concludes with a section which describes the organization of the report. In this section, the content of each of the remaining chapters is briefly summarized.

Goals

The primary goal of this study is to determine whether the KSU Adhoc Method of Database Design produces a schema in Domain Key Normal Form. As stated above, Domain Key Normal Form is the highest level of schema design and as a result, Domain Key is the goal of most designers. If it can be shown that the KSU Adhoc Method produces a Domain Key Normal Form schema, then this study will open the door for further work to determine if this adhoc method could serve as a mechanical model to transform lower normal form schemas into Domain Key schemas. A mechanical model of this type does not currently exist.

A second goal of this study is to design and implement an automated database system in support of a real world database requirement. The implementation must consist of a functional system which satisfies all user requirements and it must be delivered with full documentation.

Problems Solved

This study achieved both the goals listed in the above section. The KSU Adhoc Method of Design was utilized in a real world situation and the resulting schema was compared with the criteria for Domain Key Normal Form. The result of the comparison indicated that the schema obtained from the KSU Adhoc Method was in Domain Key Normal Form.

The study also produced a functionally automated database system which was designed and implemented to support the KSU Army ROTC Department's recruiting activities. A fully functional system was delivered to the department with complete documentation. The system was implemented using available software and it was compatible with other automated systems utilized within the department.

Organization of the Report

This report is organized into four chapters and is supported by numerous appendices.

Chapter 2, Relevant Research, addresses the subject of database design theory. It begins with a brief description of various design models/tools which are used by designers to guide them through the design process. Next, the subject of normal forms is addressed. Each level of normal form, (First (1NF), Second (2NF), Third (3NF), Fourth (4NF), Fifth/Projection-Join (5NF/PJFN), and Domain Key (DKNF)), is defined, and the anomalies associated with each level are described. Anomalies are undesirable properties in a database which have unpredictable effects on data during update, insertion, or deletion activities. In general, the problems created by anomalies are reduced as the level of

normal form increases. Domain Key Normal Form, the highest level of normal form, is virtually free of anomalies.

Chapter 2 also describes the KSU Adhoc Method of Design and addresses the main question of whether the KSU Adhoc Method produces a Domain Key Normal Form schema. An example of a schema produced by the KSU Adhoc Design Method is compared with the criteria for Domain Key Normal Form and the results are discussed in the chapter's conclusion.

Chapter 3, Application Environment, addresses the application environment for this study. It describes the critical process of identifying manual database systems as possible candidate systems for automation. This was a critical task in this study since the selected system had to be complex enough to satisfy the design requirements of this study while at the same time be simple enough to implement within the available timeframe.

The KSU Army ROTC department offered several potential candidates for automation. The chapter describes the organization of the department and addresses each candidate system in detail. Hardware and software constraints on the design are also identified. After considering the advantages and disadvantages of each candidate system, the department's recruiting system was selected to support this study.

Chapter 4, Application System Design and Implementation, addresses the actual design and

implementation of the automated recruiting system. Each step in the design process is described in detail, along with the rationale for the decisions made throughout the design. The term "suspense date" as used in this chapter, refers to a specific date in the future on which the user must perform a specified action. For example, if the user is required to send a letter to a prospective student in six months (Nov 2), then November 2 is the suspense date.

The implementation of the design called the PROSPECT System is also addressed in chapter 4. The function of each program is described in detail, and the reader is referred to the appropriate appendix for the actual program code.

Chapter 5 is the final chapter of the report. It addresses the conclusions of the study and areas of future work. The main conclusion is that the KSU Adhoc Method of Design will produce a Domain Key Normal Form schema. This is significant since it opens the door for future study and research in this area. The eventual goal of such study should be to develop a mechanical method to transform a schema from a lower normal form into Domain Key Normal Form. Other future work described in the chapter involves enhancements to the PROSPECT system and potential spinoff systems which may be beneficial in other functional areas.

Chapter 2

Relevant Research

Introduction

Database design is as much an art as it is a science. While there are a number of design tools/models available to aid the designer in his task, the effectiveness of the design is largely based on the experience and intuitive ability of the designer to 1) accurately evaluate the current system; 2) identify alternatives to satisfy both the short and long term database requirements of the user; 3) integrate hardware and software constraints; 4) select the most feasible alternative; and 5) implement the system.

This chapter addresses the subject of database design and briefly discusses some of the models available to the designer. The concept of normalization is also reviewed and the impact of higher normal forms on the database is discussed. The main focus of the chapter, however, is a discussion of Domain Key normal form and the KSU adhoc method of design. This chapter suggests the KSU adhoc method of design as a way to achieve Domain Key normal form.

Design

There are many design models available to the database designer. These models provide a guide or outline for the designer throughout the design process.

Kroenke [1], suggests a four stage approach to the design problem: stage 1, Specify Requirements; stage 2, Evaluate Alternatives; stage 3, Design; and, stage 4, Implementation. Each stage is further divided into a series of phases or steps which guide the designer through the entire development process, ensuring that the designer leaves no stone unturned throughout his journey.

Dr. Elizabeth Unger [2] suggests a seven step design life cycle consisting of: 1, Predesign Evaluation; 2, Information Modeling; 3, Logical database design; 4, DBMS selection; 5, Cost/benefit analysis; and 6, Physical Design/Implementation; 7, Test and Evaluate. This is a more thorough approach, and each step results in a product or products which are used to document the design process and aid in the decision process throughout the development.

Dr. Paul Fisher [3] suggests a detailed analysis of input and output documents as a reliable tool for determining the essential data elements in a system, and separating modifiable from nonmodifiable data. This was step two of a ten step method he suggested for system analysis in a network environment.

Since each design environment is unique, a particular design model cannot be universally applied to every design. Designers normally select a specific model and modify it to adapt to the specific design environment. For this study, elements from each of the models were used to create a unique model for this design. The final model consisted of the following steps: 1, Define the Problem; 2, Evaluate the current system; 3, Identify essential/nonessential data (document analysis); 4, Determine functional/multivalued dependencies; 5, Define a schema of the highest possible normal form using synthesis and decomposition techniques; 6, Implement the design.

Normal Forms

Relational database theory contends that not all database designs are equally good because of the existence of modification anomalies. These modification (insertion/deletion/update) anomalies are present to some degree in most databases, but they can be eliminated by elevating the database to a higher normal form.

There are at least seven accepted levels of normal forms. Each successive level incorporates the aspects of all lower levels. The levels are: First Normal Form, Second Normal Form, Third Normal Form, Boyce-Codd Normal

Form, Fourth Normal Form, Fifth Normal Form, and Domain Key Normal Form.

First Normal Form (1NF): A relation is in first normal form if the domains of all attributes of the relation are atomic, i.e. indivisible units [4]. Essentially, all relations are in 1NF as long as there are no repeating groups in the relation [5]. Relations which are in first normal form possess modification anomalies.

Second Normal Form (2NF): A relation is in second normal form if it is in 1NF and every nonprime attribute is fully dependent on every key. This can be achieved by eliminating extraneous attributes from relations thereby preventing the occurrence of partial dependencies. Relations of 2NF also possess modification anomalies and therefore, it is not normally sufficient for database design.

Third Normal Form (3NF): A relation is in third normal form if it is in 2NF and no nonprime attributes are transitively dependent on any key. 3NF may contain anomalies, but it is considered an acceptable level of normalization for most database designs. A 3NF schema can always be found which ensures preservation of dependencies and lossless-join decomposition [6].

Boyce-Codd Normal Form (BCNF): A relation is in BCNF if the left hand side of every dependency from the closure that applies to the relation is a key. A schema in BCNF has no anomalies regarding functional dependencies, but it does not consider multivalued dependencies. Additionally, BCNF decompositions do not guarantee a lossless join and they are not dependency-preserving.

Fourth Normal Form (4NF): A relation is in 4NF if it is in BCNF and all multivalued dependencies in the relation are in fact functional dependencies, i.e., all multivalued dependencies in the relation are trivial. The major problem with 4NF is that we cannot ensure dependency preservation.

Fifth Normal Form (5NF): Fifth normal form is also referred to as Projection-Join Normal Form (PJ/NF). A relation is in 5NF if and only if every join dependency in the relation is a consequence of the candidate keys of the relation [7]. There are no straight forward "real world" interpretations for join dependencies, and therefore, discovering all the join dependencies is a nontrivial problem [2]. Every multivalued dependency is considered a join dependency and therefore every PJ-NF is also in 4NF. As a result, decompositions into 5NF may not be dependency preserving.

Domain-Key Normal Form (DKNF): DKNF is considered the highest possible normal form, and it is therefore the objective of many designers. Relations in DKNF contain no modification anomalies. It is based on three notions: 1) each attribute value is a subset of a domain; 2) the declared key is a super key of the relation and 3) a general constraint is expressed as a predicate on the set of all relations on a given scheme. Modification anomalies are eliminated in DKNF by two fundamental processes: 1) enforcing domain definitions via editing and 2) requiring key attributes to be unique. A relation schema is in DKNF if every constraint can be inferred by simply knowing the set of attribute names and their underlying domains, along with the set of keys [8].

A particular schema can be transformed via decomposition or synthesis to achieve a higher normal form. However, the decomposition or synthesis can result in a lossy or unfaithful schema which does not satisfy the original design requirements. The KSU adhoc design method presents a practical design life cycle method for achieving Domain Key Normal Form by augmenting familiar decomposition and synthesis techniques with one additional step. This modified method was used in this study.

KSU adhoc design method

The KSU adhoc design method was developed to integrate multivalued dependencies into a schema without losing

functional dependencies. Traditional decomposition techniques often result in the loss of critical functional dependencies when multivalued dependencies are decomposed into 4NF. When this occurs, the resulting schema no longer satisfies the design requirements. This creates a dilemma for the designer of how to incorporate the MVDs into the schema while achieving the highest possible normal form.

In order to achieve a 4NF schema, MVDs must be trivial or the left hand side of each nontrivial MVD must be a super key. As an example, Figure 2.1 provides a list of FDs and MVDs identified for the database design in this study. Figure 2.2 illustrates what happens to the original MVDs when traditional decomposition techniques are used to decompose the relation into 4NF. Note that all the FDs with Last_name, First_name, and Middle_Initial on the left hand side were lost. Since the combination of these three fields determines the majority of the FDs in the system, it is senseless to continue the decomposition further because the resulting schema will not satisfy the design requirements. The designers dilemma is to find a way to incorporate the critical MVDs into the schema and achieve the highest normal form without losing critical dependencies.

Dependencies

FUNCTIONAL DEPENDENCIES (FDs)

LAST-NAME, FIRST-NAME, MI > ADMITTED, AREA-CODE, CADET-REP,
CITY, CADRE-REP, COLLEGE-CREDITS, COL-GPA,
COMMENTS, DOB, ENTRY-DATE, DISPOSITION,
DISPOSITION-DATE, NEXT-FOLLOWUP-TYPE, NEXT-
FOLLOWUP-DATE, HIGH-SCHOOL, HS-GRAD-YR, SCHOOL-
DIST, IN-GOLD-QUEST, LAST-CONTACT-DATE, MAJOR,
MILITARY-EXPERIENCE, PHONE, REFERRED-BY,
REGION, SCHOLARSHIP-APP, SEX, SOURCE, SSN,
STATE, STREET, TITLE, ZIP, RACE, INFLUENCER;

LAST-NAME, FIRST-NAME, MI, CONTACT-DATE, CONTACT-METHOD ->
CONTACT-REP, CONTACT-REMARKS:

MULTIVALUED DEPENDENCIES (MVDs)

CONTACT-REP, CONTACT-DATE --> LAST-NAME, FIRST-NAME, MI

CADET-REP --> LAST-NAME, FIRST-NAME, MI

CADRE-REP --> LAST-NAME, FIRST-NAME, MI

LAST-NAME, FIRST-NAME, MI --> FOLLOW-UP-TYPE, FOLLOW-UP-
DATE, FOLLOW-UP-COMPLETED

LAST-NAME, FIRST-NAME, MI --> CONTACT-REP

FOLLOW-UP-TYPE, FOLLOW-UP-DATE --> LAST-NAME, FIRST-NAME,
MI

Figure 2.1

Traditional Decomposition

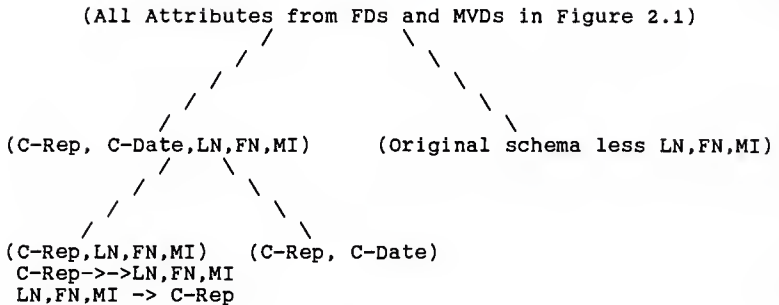


Figure 2.2

The KSU adhoc method provides a solution to this dilemma. The method is simple and is outlined in Figure 2.3.

KSU Adhoc Design Method

- Identify the Functional and Multivalued Dependencies.
- Decompose the Functional dependencies into 3NF using traditional decomposition or synthesis techniques.
- Create a separate relation for each multivalued dependency.
- Define the combination of all the attributes in each multivalued relation as the key to the relation.

Figure 2.3

Once the FDs and MVDs are identified, traditional decomposition/synthesis techniques are used on the FDs to produce a schema in 3NF. Figure 2.4 illustrates the 3NF schema which resulted from running the FDs in Figure 2.1 through an automated version of Bernstein's synthesis algorithm 2 [11].

THE SCHEMA IN 3NF (FDs only)

```
(MI FIRST-NAME LAST-NAME ) > ADMITTED AREA-CODE CADET-REP
CITY CADRE-REP COLLEGE-CREDITS COL-GPA COMMENTS
DOB ENTRY-DATE DISPOSITION DISPOSITION-DATE
NEXT-FOL-UP-DATE NEXT-FOL-UP-TYPE HIGH-SCHOOL
HS-GRAD-YR LAST-CONTACT-DATE SCHOOL-DIST IN-
GOLD-QUEST MAJOR MIL-EXPERIENCE PHONE REFERRED-
BY REGION SCHOLARSHIP-APP SEX SOURCE SSN
STATE STREET TITLE ZIP RACE INFLUENCER

(CONTACT-METHOD CONTACT-DATE MI FIRST-NAME LAST-NAME ) >
CONTACT-REP CONTACT-REMARKS
```

Figure 2.4

The MVDs from Figure 2.1 are incorporated by simply creating a separate relation for each MVD. The key to the relation is a combination of all attributes in the relation. The

resulting relations and their keys are shown in Figure 2.5. The question remains, "Is the resulting schema in Domain Key Normal Form?"

Relations from the KSU Adhoc Design Method

CONTACT (Last_Name, First_Name, Middle_Initial, Contact_Date, Contact_Method, Contact_Remarks, Contact_Rep)

Key: Last_Name, First_Name, Middle_Initial, Contact_Date, Contact_Method

PROSPECT (Last_Name, First_Name, Middle_Initial, Admitted, Area_Code, Cadet_Rep, City, Cadre_Rep, College_Credits, Col_GPA, Comments, DOB, Entry_Date, Disposition, Disposition_Date, Next_FollowUp_Type, Next_FollowUp_Date, High_School, HS_Grad_Year, School_Dist, In_GoldQuest, Last_Contact_Date, Major, Military_Experience, Phone, Referred_By, Region, Scholarship_App, Sex, Source, SSN, State, Street, Title, Zip, Race, Influencer)

Key: Last_Name, First_Name, Middle_Initial

M_CONTACT (Contact_Rep, Contact_Date, Last_Name, First_Name, Middle_Initial)

Key: Contact_Rep, Contact_Date, Last_Name, First_Name, Middle_Initial

M_CADET (Cadet_Rep, Last_Name, First_Name, Middle_Initial)

Key: Cadet_Rep, Last_Name, First_Name, Middle_Initial

M_CADRE (Cadre_Rep, Last_Name, First_Name, Middle_Initial)

Key: Cadre_Rep, Last_Name, First_Name, Middle_Initial

M_FOLLOW (Last_Name, First_Name, Middle_Initial, Follow_up_Type, Follow_up_Date, Follow_up_Completed)

Key: Last_Name, First_Name, Middle_Initial, Follow_up_Type, Follow_up_Date, Follow_up_Completed

M_NAME (Last_Name, First_Name, Middle_Initial, Contact_Rep)

Key: Last_Name, First_Name, Middle_Initial, Contact_Rep

M_FOL_NAME (Follow_Up_Type, Follow_Up_Date, Last_Name, First_Name, Middle_Initial)

Key: Follow_Up_Type, Follow_Up_Date, Last_Name, First_Name, Middle_Initial

Figure 2.5

Domain Key Example

There is no way to convert a relation into to DKNF automatically [9]. To determine if the above schema is in DKNF, we must incorporate the data elements in Figure 2.1 so that constraints are logical consequences of domains and keys. If that cannot be achieved, the constraints must be built into application programs which process the database.

Figure 2.6 provides a list of domain definitions from the above example. The constraints in the domain are represented by the FDs and MVDs listed in Figure 2.1. By creating a separate relation for each multivalued dependency, as was done in the KSU Adhoc Method, the constraints become logical consequences of the domains and keys. Therefore, while it may not be the optimum solution, the schema produced by the KSU adhoc method in Figure 2.2 is a valid DKNF schema.

Domain Definitions

Admitted	IN	{A,P,N, blank}
Area_Code	IN	Integer(3) may be blank
Cadet_Rep	IN	Char(10)
Cadre_Rep	IN	Char(10)
City	IN	Char(20)
College_Credits	IN	Integer(3) may be blank
Col_GPA	IN	Real(3) 9.99 may be blank
Comments	IN	Char(250)
Contact_Date	IN	Date(8) MM/DD/YY
Contact_Method	IN	Char(1) {P,T,M,N}
Contact_Rep	IN	Char(10)
Contact_Remarks	IN	Char(30)
Date_of_Birth	IN	Date(8) MM/DD/YY
Disposition	IN	Char(2) {EN,IN,NI,NQ,OO}
Disposition_Date	IN	Date(8) MM/DD/YY
Entry_Date	IN	Date(8) MM/DD/YY
First_Name	IN	Char(10)
Follow_up_Complete	IN	Char(1) {Y,N}
Follow_up_Type	IN	Char(20)
Follow_up_Date	IN	Date(8) YY/MM/DD
High_School	IN	Char(30)
HS_Grad_Year	IN	Integer(2)
In_Gold_Quest	IN	Char(1) {Y,N}
Influencer	IN	Char(20)
Last_Name	IN	Char(20)
Last_Contact_Date	IN	Date(8) MM/DD/YY
Major	IN	Char(20)
Middle_Initial	IN	Char(2)
Military_Experience	IN	Char(4) {AA-C,AA-P,AR-C,AR-P,NG-C,NG-P,USNC,USNP,MC-C,MC-P,AF-C,AF-P,OTHR,blanks}
Next_Follow_up_Date	IN	Date(8) MM/DD/YY
Next_Follow_up_Type	IN	Char(20)
Phone	IN	Integer(8) ###-####
Race	IN	Char(1) {C,N,R,M,X,Z,}
Referred_By	IN	Char(20)
Region	IN	Integer(1) {0..9,blank}
Scholarship_App	IN	Char(1) {4,3,2,A,N,O}
School_District	IN	Integer(3)
Sex	IN	Char(1) {M,F}
Source	IN	Char(2) {IN,GQ,UR,GG,GO,OO}
SSN	IN	Integer(9) may be blank
State	IN	Char(2)
Street	IN	Char(20)
Title	IN	Char(4) {Mr,Mrs,Miss,Ms}
Zip	IN	Integer(9) #####-####

Figure 2.6

Conclusion

One goal of all database designers is to design a database with minimal anomalies. Fagin proved that Domain Key Normal Form is virtually free of anomalies [8]. Therefore, DKNF should be a goal of all database designers.

The KSU Adhoc Design Method offers an alternative to the traditional decomposition method for achieving a higher normal form schema. This method formalizes a standard field practice of creating a separate relation for each MVD. The schema which results from this design method not only preserves dependencies, but also meets the requirements of Domain Key Normal Form.

Chapter 3

Application Environment

Introduction

Selection of an appropriate application environment was axiomatic to the success of the study. The environment had to be complex. It had to contain multivalued dependencies and exhibit all of the problems and decision points necessary to explore each design step and test the full range of design techniques. However, the environment also had to be limited in scope to ensure that at least a partial implementation of the design was feasible within acceptable time constraints.

The KSU Army ROTC department was targeted as a potential application environment for the study. With the department's concurrence, a preliminary review of their automated database requirements was conducted. The result of the review indicated that the department's application environment met the complexity and scope requirements of this study. Additionally, it provided alternatives for a specific design selection.

Characteristics of the Environment

The KSU Army ROTC department recruits and trains college students in military science subjects in preparation

for a commission in the active army, the army reserve and the national guard. The department is staffed with six officers, 3 non-commissioned officers, a secretary and a records clerk. The primary mission of the department is to train approximately 110 to 125 students (cadets) annually and commission them as second lieutenants upon graduation. The students are divided among four classes, MSI (freshman), MSII (sophomores), MSIII (juniors), and MSIV (seniors). One training officer from the department is assigned to each class to monitor the cadets' progress and ensure each cadet meets all requirements for commissioning.

The activities of the department are actively supervised by a regional headquarters. Numerous forms and reports are required by the regional headquarters on a regular basis to monitor cadet management and recruiting actions. To assist with the management and reporting functions, the department has two personal computers: a ZENITH 158 PC and, a WANG PC both with dual 5.25 floppy disk drives and a WANG VS5 computer.

The Zenith 158 PC is a standalone single user system. Department training officers have implemented a small cadet management database system on the Zenith using dBASE III PLUS software to maintain local cadet management data. The WANG PC can be used as a stand alone computer, but it is also connected to a WANG VS5 computer with a 70 meg hard drive and a 2400 baud modem. The WANG system was installed

in the department by the regional headquarters to provide a data link for future automation of reporting requirements. At present, the WANG system serves primarily as a remote data entry device and little information is available to the department regarding the eventual capability of the system. It appears that the WANG system and the regional database will eventually provide the department with access to some automated cadet management data, but it is doubtful that the regional database will ever satisfy all of the local information requirements of the KSU department.

The significance of the WANG system and the local database development activity is that these developments place a constraint on database design initiatives for the ROTC department. Any new design must be implemented as a single user system, and it must run on both the WANG and the Zenith computers. Additionally, a new design must blend with local database design initiative and be easily used and maintained by non-technical users.

Potential for Database application within the Department

Four potential areas for database design were identified within the KSU Army ROTC Department: cadet management, recruitment, supply, and budget. Each area was evaluated to determine whether its complexity and scope would satisfy the goals of this study. If more than one

area satisfied the study goals, the ROTC department's priority for design was considered.

A review of the supply and budget operations within the department quickly revealed that these operations were trivial. The low number of transactions and minimum reporting requirements did not justify automation of these areas. Additionally both areas lacked the complexity required to satisfy the objectives of this study. As a result, they were eliminated as candidates for design. However, cadet management and recruitment activities remained justifiable candidates, and a detailed discussion of each activity is provided below:

Cadet Management.

Cadet management involves maintaining data and providing reports regarding cadet academic and enrollment information. The type of data maintained on each cadet is as follows:

- a. Personal Data
- b. Enrollment Data
- c. Scholarship Data
- d. Basic Course Data
- e. Prior Active / Reserve Military Training Data
- f. Prior ROTC Training Data

- g. Placement Credit Data
- h. Academic Subjects Validated for Credit
- i. Compression of Military Science Data
- j. Height / Weight Data
- k. Physical Readiness Test Data
- l. Medical Examination Data
- m. Selection Criteria Data
- n. Contracts and Related Forms Data
- o. Summer Camp / Special Training Data
- p. Academic / ROTC Probation Data
- q. Scholarship Data
- r. Disenrollment Data
- s. Graduation Data
- t. Academic Data
- u. Military Science Data
- v. Subsistence Data
- w. Security Investigation Data
- x. Appointment Data
- y. Active Duty / Reserve Forces Duty Data
- z. Certificates / Awards Data

The above data is maintained manually on a TRADOC Form 476. This form is the corner stone for cadet management data, and it is an item of inspection during visits by regional headquarters personnel.

Reports from this manual database vary in type and frequency. Each report request, whether internally or

externally generated requires a manual scan of the TRADOC Forms 476 by the records clerk to compile and consolidate the requested data.

In an attempt to speed up this manual process, a parallel automated database system was been designed by a department training officer using the Zenith 158 Personal Computer and dBASE III PLUS software. The system consists of 12 database files and generates approximately 12 reports. The reports consist of rosters, periodic lists, and management / summary data which provide timely information to training officers and the KSU Professor of Military Science (PMS). The automated database is, in essence, a reduced version of the manual database. Not all of the data elements on the form 476 are represented in the automated database, but those, which are represented, duplicate data elements in the manual database. This situation leads to an obvious data consistency problem, especially since the two databases are maintained by different personnel and are updated on different dates. Frequent coordination between the training officers and the record clerk is required to reduce the effect of this problem and maintain acceptable consistency between the two databases.

The current automated system is operating effectively and satisfies most requirements of the department. However, enhancements are desired to add data elements for height, weight, physical training, and KSU academic information.

The recent installation of the WANG VS5 offers some hope for a consolidated automated cadet management system in the near future. The system is scheduled to be fully operational by the start of the fall semester 1988, but the overall capability of the system remains uncertain. KSU ROTC department personnel remain skeptical regarding the WANG's ability to satisfy local reporting and query requirements.

Recruitment

A second and equally important mission for the ROTC department is the recruitment of qualified students and potential students for the Army ROTC program. One officer in the department has overall responsibility for the recruitment program, and it is his primary responsibility to maintain data on prospective cadets and assign cadre and cadet representatives to each qualified prospect.

Information on prospective cadets is obtained in four ways: 1) internally, through high school visits and information booths on campus; 2) referrals from the national Gold Quest center; 3) Goldminer team referrals; and 4) US Army Recruiting Command referrals. A fifth source of prospects may become available in the near future through an agreement with the KSU recruitment office regarding an exchange of information.

Internal recruiting is conducted on campus and through high school visits. Personal information about a prospective student is obtained from interested personnel and recorded on a local recruiting form. If the prospect is qualified for the program, an action officer (cadre representative) is assigned to follow up the contact with additional information about the program. The personal information on the prospect is maintained on file as long as the prospect remains qualified for the program.

Information obtained through internal recruiting activities may also be forwarded to the national Gold Quest system. The national Gold Quest system provides an automatic suspense system to disperse scholarship information or other data on specified dates and remind local cadre representatives of follow up correspondence. This procedure is usually reserved for interested high school students who are several months from graduation.

The national Gold Quest system also forwards data on prospective KSU students to the ROTC Department. This "referral action" is accomplished via a post card which contains personal data on the prospective cadet. Upon receipt of this information, a local cadre representative is assigned to the prospect, and a record of contacts with the prospect is maintained on the reverse side of the form. Final disposition of these cases, i.e., enrolled, not-

interested, not qualified, etc., is reported by the department to the Gold Quest system.

Goldminer Team referrals are forwarded to the department by special regional recruiting teams called Goldminers. These referrals are handled in the same manner as Gold Quest referrals; a record of contact is maintained and the final disposition of the prospect is passed to the referring agency.

The US Army Recruiting Command also provides referrals to the ROTC department. These referrals are accompanied by three self addressed post cards which are returned at various stages of contact with the prospect. The first card is returned immediately to acknowledge receipt of the referral. The second card is returned to report the results of initial contact with the referral. The third card is returned to report the final disposition of the case.

To accomplish its recruitment mission and satisfy the reporting requirements of the referring agencies, the department maintains the following data on each cadet:

- Date of contact
- Title
- Name
- Address
- Telephone number
- Source of referral

- Date of Birth
- High School
- Graduation Date
- College Credits
- College GPA
- Prior military service data
- Assigned Cadre Representative
- Assigned Cadet Representative
- Name of Sponsor or Influence
- Record of Contacts
- Recruitment status
- Comments
- Final Disposition
- Suspense Dates for Follow-up Actions

Until recently, the ROTC department relied solely on a manual system for maintaining the above data. However, a rudimentary database system was developed on the Zenith 158 personal computer using dBASE III PLUS software to meet local management and reporting requirements. The automated system was hurriedly designed and is only marginally effective in solving the needs of the department. The program would benefit greatly from a newly designed automated database system.

Conclusion

Both the cadet management and recruiting systems provide the multivalued dependencies and complexity necessary to meet the goals of this study. Additionally, the scope of each application is within the limit of the study, thereby ensuring at least a partial implementation of the design within available time constraints. However, the design of an automated database in support of recruiting activities was given priority by the ROTC department for three reasons: 1, the local automated database for cadet management is adequately meeting the department's needs; 2, the WANG system provides a potential near-term solution to the cadet management problem and therefore, any design effort on that system may be obsolete before it is fully implemented; and 3, the current automated database in support of recruitment is not adequately meeting the needs of the department.

Based on the above rationale, the recruitment system was selected for design in support of this study.

Chapter 4

Application System Design and Implementation

Introduction

This chapter addresses the actual design and implementation of the automated data base for the KSU Army ROTC Department's recruiting mission. The chapter is divided into six sections. Section one, The Manual System, describes the manual system which was used by the ROTC department prior to implementing the automated design. Section two, Analysis, addresses the analysis phase of the design and describes the products and decisions resulting from this phase. Section three, The DBMS, discusses the rationale for selecting the specific DBMS software. Section four, The Implementation, describes the final design and addresses the decisions which impacted on the design. Section five, Programs, describes each problem developed for the new PROSPECT system. The last section, Conclusion, discusses the impact of the system on the user, and user satisfaction with the system.

The Manual System

In the past, the Army ROTC department relied primarily on a manual system to support its recruiting mission. Biographic information on prospective cadets was obtained from personal interviews or from referrals by other agencies such as the Goldminer Teams, the US Army Recruiting Command, or the Gold Quest System.

This information was recorded on a data sheet (Figure 4.1). The completed data sheets were given to cadre representatives who were assigned responsibility for contacting, recruiting, and providing information to the prospective cadets. Each contact between a cadre representative and the prospective cadets was recorded on the data sheets. Follow-up actions and suspense dates were recorded in the comments section of the form.

The data sheets were retained on file as long as the prospect remained a candidate for the ROTC program. Information from the sheets was used to provide updates to the local commander and referring agencies regarding potential enrollment and the status of the recruiting effort.

This manual database system was not responsive to the management needs of the department. Data was maintained on over 300 prospective cadets, but the database was divided among four or five Cadre representatives. As a result, the officer charged with overall responsibility for the recruitment program actually had very little control of the data. He found it difficult to respond to queries from the local commander or the regional office without personally obtaining data from the cadre representatives and manually compiling a report. The manual system also made it difficult to determine what prospects were assigned to which Cadre representatives. As a result, the recruitment

officer had difficulty ensuring equitable distribution of prospects to cadre.

The Analysis

The first step in the design process was to conduct a forms analysis. The objective of this analysis was to determine essential data, extraneous/nonessential data, and identify which elements, if any, could be derived from data in the system. The personal data sheet (Figure 4.1) was of course the primary focus of this analysis. However, additional forms from the US Army Recruiting Command, the Gold Quest System, and the Goldminer Teams were also analyzed.

The next step in the analysis was to review existing reports and determine specific requirements/expectations from the system. The review indicated that the reports produced from the manual system were primarily one-time requirements, consisting essentially of counting the number of active prospects, comparing that number with the number probable enrollees, and recording the names and number of prospects referred by various sources. All of these were, for the most part, nonrecurring one-time requests. In every case, the selection criteria for computations was different. As a result, these reports were used only as a guide to develop a system with the flexibility to respond effectively to a variety of reporting requirements.

Data Sheet

Status (pencil): HOT WARM COLD _____ Eligibility Checklist
 date

KANSAS STATE UNIVERSITY
 ARMY ROTC LEADS TRACKING SYSTEM

Date: _____ Prospect Influencer

Title: Mr. Mrs. Miss Ms. _____

Name: _____
 Last First MI

Street: _____

Town: _____

State: KS MO OK ME CO _____ Zip _____

Area Code: _____ Phone: _____

In LEADS: YES NO Green-to-Gold Source: _____

Age: _____ DOB: _____ ACT/SAT: _____

High School: _____ Grad Date: _____ GPA _____

University: KSU KU WSU PSU _____

Major: _____ GPA: _____ Credits: _____

AD VET AR NG GT: _____ ETS: _____ TIS/BASD: _____ Unit: _____

Comments: _____

Contact Date	ROTC Rep	Method	Remarks
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____

Disposition: Enrolled _____ Not Interested Ineligible _____

The ROTC department developed a list of recurring reports which were desired from the new system to assist in the recruitment management function. This list (Figure 4.2) served as the primary guide for system design requirements.

As a result of the above analysis, essential data elements were identified and a data dictionary was developed. This dictionary was continually revised throughout the design process as decisions regarding the role and structure of each data element was determined. The final dictionary (Appendix D) contains 47 data elements.

Once the essential data elements were identified, the important task of determining functional and multivalued dependencies among the data elements was performed. This was a critical step in the design phase since decisions regarding dependencies greatly influence the final structure of a database. Mistakes and oversights during this process were costly in terms of time and effort. When dependencies were incorrectly identified or omitted, all synthesis and decomposition techniques had to be recomputed to determine the impact on the final schema. This was an important lesson learned from this study.

The final set of functional and multivalued dependencies are shown in Figure 4.3. The functional dependencies were run through an automated version of Bernstein's synthesis algorithm 2 [11] which eliminated extraneous data elements and redundant functional

dependencies and produced a schema in 3rd normal form. Since the algorithm does not consider multivalued dependencies, the "KSU adhoc method" was used to separate the multivalued dependencies into separate schemas as shown in Figure 4.4. These schemas became the initial bases for the final design.

The DBMS

The ROTC department had recently purchased dBase III Plus software, and personnel within the department were familiar with its use. They had also developed some limited programming expertise with the software, and therefore it was strongly encouraged that any new design be implemented using dBase III Plus.

One concern regarding the use of dBase III Plus was that it was purchased to operate on the Zenith system only. It would not run on the WANG system, and no funds were available to purchase software for the WANG. This restriction was a major concern since it would restricted operation of the new system to the Zenith only, and access to the system would be limited. Additionally, since the new design did not utilize the latest hardware available to the department, the lifetime of the system was limited. This concern was alleviated, however, when the regional headquarters provided the department with IBM PC emulation packages for the WANGs.

ROTC Report Requirements

Priority	Report
4	- List of Gold Quest Prospects by name in alphabetical order.
1	- List of prospects by Cadre member.
2	- List of prospects not contacted within past 30 days.
2	- List of prospects not contacted within past 60 days.
2	- List of prospects not contacted within past 90 days.
2	- List of prospects with a HS grad date of 88.
2	- List of prospects with a HS grad date of 89.
2	- List of prospects with a HS grad date of 90.
2	- List of prospects with a HS grad date beyond 90.
3	- List of all Green to Gold Prospects/Referrals.
4	- List of all prospects with any college credits.
2	- List of all Kansas Prospects.
2	- List of all prospects by Region in Kansas.
4	- List of all USAREC Referrals.
4	- List of all Goldminer Team Referrals.
3	- List of all 4-Year scholarship applicants.
2	- List of all prospects by high school (school district).
3	- List of all prospects by city (zip code).
3	- List of all Green to Gold referrals by month (or past 30 days).
1	- Address Labels.
4	- Summary of prospect sources.
2	- List of all Scholarship Recipients by type.
1	- Individual Query

Priority: 1 = High, 4 = Low

Figure 4.2

Dependencies

FUNCTIONAL DEPENDENCIES (FDs)

LAST-NAME, FIRST-NAME, MI -> ADMITTED, AREA-CODE, CADET-REP, CITY, CADRE-REP, COLLEGE-CREDITS, COL-GPA, COMMENTS, DOB, ENTRY-DATE, DISPOSITION, DISPOSITION-DATE, NEXT-FOLLOWUP-TYPE, NEXT-FOLLOWUP-DATE, HIGH-SCHOOL, HS-GRAD-YR, SCHOOL-DIST, IN-GOLD-QUEST, MAJOR, MILITARY-EXPERIENCE, PHONE, REFERRED-BY, REGION, SCHOLARSHIP-APP, SEX, SOURCE, SSN, STATE, STREET, TITLE, ZIP, RACE, INFLUENCER;

LAST-NAME, FIRST-NAME, MI, CONTACT-DATE, CONTACT-METHOD -> CONTACT-REP, CONTACT-REMARKS;

MULTIVALUED DEPENDENCIES (MVDs)

CONTACT-REP, CONTACT-DATE ->> LAST-NAME, FIRST-NAME, MI

CADET-REP ->> LAST-NAME, FIRST-NAME, MI

CADRE-REP ->> LAST-NAME, FIRST-NAME, MI

LAST-NAME, FIRST-NAME, MI ->> FOLLOW-UP-TYPE, FOLLOW-UP-DATE, FOLLOW-UP-COMPLETED

LAST-NAME, FIRST-NAME, MI ->> CONTACT-REP

FOLLOW-UP-TYPE, FOLLOW-UP-DATE ->> LAST-NAME, FIRST-NAME, MI

Figure 4.3

Initial Schemas

CONTACT (Last_Name, First_Name, Middle_Initial, Contact_Date,
Contact_Method, Contact_Remarks, Contact_Rep)

Key: Last_Name, First_Name, Middle_Initial, Contact_Date,
Contact_Method

PROSPECT (Last_Name, First_Name, Middle_Initial, Admitted,
Area_Code, Cadet_Rep, City, Cadre_Rep, College_
Credits, Col_GPA, Comments, DOB, Entry_Date,
Disposition, Disposition_Date, Next_FollowUp_
Type, Next_FollowUp_Date, High_School, HS_Grad_
Year, School_Dist, In_GoldQuest, Major, Military_
Experience, Phone, Referred_By, Region,
Scholarship_App, Sex_Source, SSN, State, Street,
Title, Zip, Race, Influencer)

Key: Last_Name, First_Name, Middle_Initial

M_CONTACT (Contact_Rep, Contact_Date, Last_Name, First_Name,
Middle_Initial)

Key: Contact_Rep, Contact_Date, Last_Name, First_Name,
Middle_Initial

M_CADET (Cadet_Rep, Last_Name, First_Name, Middle_Initial)

Key: Cadet_Rep, Last_Name, First_Name, Middle_Initial

M_CADRE (Cadre_Rep, Last_Name, First_Name, Middle_Initial)

Key: Cadre_Rep, Last_Name, First_Name, Middle_Initial

M_FOLLOW (Last_Name, First_Name, Middle_Initial, Follow_up_
Type, Follow_up_Date, Follow_up_Completed)

Key: Last_Name, First_Name, Middle_Initial, Follow_up_Type,
Follow_up_Date, Follow_up_Completed

M_NAME (Last_Name, First_Name, Middle_Initial, Contact_Rep)

Key: Last_Name, First_Name, Middle_Initial, Contact_Rep

M_FOL_NAME (Follow_Up_Type, Follow_Up_Date, Last_Name,
First_Name, Middle_Initial)

Key: Follow_Up_Type, Follow_Up_Date, Last_Name, First_Name,
Middle_Initial

Figure 4.4

The Implementation

The schema produced from the "KSU adhoc method" was not an optimum Design. It contained more than the minimal number of relations and attribute occurrences. While the KSU design was efficient in terms of data organization and structure, it placed an unnecessary burden on the user who would be forced to maintain duplicate data in eight separate relations. As a result, the MVDs and FDs were reevaluated for criticality in terms of data organization and retrieval, and to determine if the constraints imposed by critical MVDs could be maintained through programming or through the enhanced capability of dBase III Plus software, in lieu of separate relations.

Based on the reevaluation of the MVDs and the problem environment, it was determined that a three relation schema would best satisfy the user requirements. Of the six relations (See Figure 4.4) formed from the MVDs (See Figure 4.3), two were maintained as one relation. The first MVD (CONTACT-REP, CONTACT-DATE ->-> LAST NAME, FIRST-NAME, MI) was dropped and an inverted index on the relation formed from the second FD was substituted. The second and third relations formed from MVDs (CADET-REP ->-> LAST-NAME, FIRST-NAME, MI and CADRE-REP ->-> LAST-NAME, FIRST-NAME, MI) were dropped from the relations and coded into software. The fourth relation formed from the MVDs was maintained but

the key was changed to LAST-NAME, FIRST-NAME, MI reducing its normal form to BCNF. The fifth relation formed from the MVD (LAST-NAME, FIRST-NAME, MI --> CONTACT-REP) was included in the relation formed from the second set of FDs. The key was changed to LAST NAME, FIRST NAME, MI and the information about Contact-Rep is determined with an index. The last relation formed from MVDs (FOLLOW-UP TYPE, FOLLOW-UP DATE --> LAST-NAME, FIRST-NAME, MI) was dropped.

The final optimum schema for this design consisted of three relations which were structured as shown in Appendix A, Appendix B, and Appendix C. The advantages of the new three relation schema were that it was a simple design which was easy to understand and it could be maintained with available expertise within the ROTC department. The disadvantages were that it forced the use of indexing techniques and programming to retrieve data and it lowered the normal form of the schema to BCNF.

The link between the three relations (called databases in dBase III Plus terminology) was the prospective cadet's name. The prospect's name as opposed to the social security number was used as a key to the records due to Privacy Act considerations and the fact that the social security number was not always available to the recruiting officers.

Three fields were created in the PROSPECT database to indicate the last contact date, next follow-up date and the next follow-up type associated with each prospective cadet.

These fields were critical to the recruitment officers management function because it provided him with information regarding suspense dates as well as a method for identifying prospects who had not been contacted for a given time period. Since the data for these fields in the PROSPECT database was maintained in the FOLLOWUP and CONTACT databases, the system was designed to automatically update these fields in the PROSPECT database each time the FOLLOWUP or CONTACT databases were updated.

The final implementation consisted of eleven programs, four screen formats, and two special report formats. The hierarchical structure of the programs and formats are displayed in figure 4.5.

Program Hierarchy

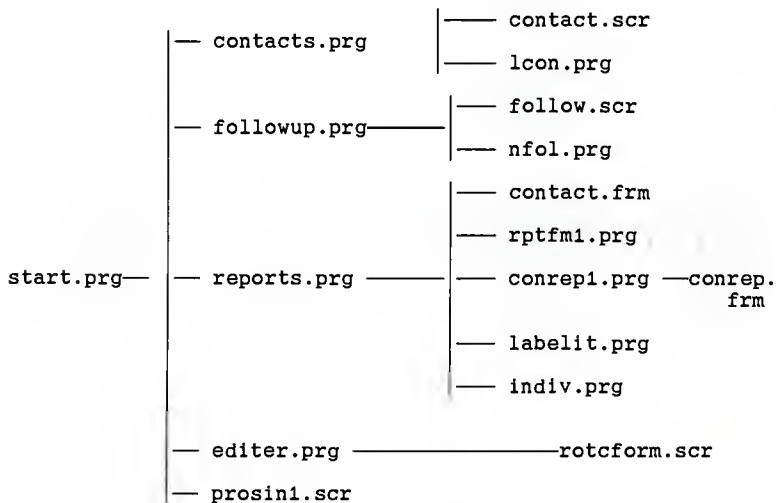


Figure 4.5
Page 4-12

The new system was designed specifically for nontechnical users. The user is guided through a series of menus and screen formats which make it extremely user friendly. In addition, a complete user's guide and documentation was provided to the department with the software. A copy of the user documentation is attached at Appendix CC.

The Programs.

START.PRG: (Source Code at Appendix R) This program displays the main menu and provides access to the PROSPECT, CONTACT, and FOLLOWUP databases. The main menu offers the user one of the seven choices (0 to 6).

0. Exits the program to the dBase Dot.
1. Allows the user to input new records. It uses a customized input format, PROSPIN1, and appends new records to the PROSPECT database.
2. Executes the program EDITER.PRG, which allows the user to edit records in the PROSPECT database.
3. Executes program REPORTS.PRG, which provides the user with a menu of predesigned reports.
4. Executes program FOLLOWUP.PRG which allows the user to input or edit data in the FOLLOWUP database.

5. Executes program CONTACTS.PRG which allows the user to input or edit data in the CONTACT database.
6. Exits the program to operating system prompt "A>".

EDITER.PRG: (Source Code at Appendix S) This program allows the user to edit records in the PROSPECT database. It prints a menu which offers four choices.

0. Exit. The user is returned to START.PRG and the main menu is displayed.
1. Allows the user to edit a specific record in the PROSPECT database. A prompt is printed on the screen asking for the last name, first name, and middle initial of the prospect, and a search is conducted using the LOCATE command. Care must be exercised by the user to ensure proper spelling of the name. If the record can not be located in the file, a message to that effect is printed and the user is returned to the edit menu.
2. Allows the user to edit all records associated with a specified cadre representative's last name. A prompt is printed on the screen asking the user to enter the last name of the cadre representative. The user keyed input is transformed into upper case letters. The PROSPECT database is then indexed by

the cadre representative field and a SEEK command is used to find the first record. As in the case above, if cadre representative's name is not in the database, a message is displayed on the screen and the user is returned to the edit menu.

3. Allows the user to edit the entire file based on an alphabetical index of the records by the prospective cadet's name.

In each case, the ROTCFORM.SCR format is used to display the records for editing.

REPORTS.PRG: (Source Code at Appendix T) This program offers the user a selection of 12 predesigned reports. Most of the reports utilize a single format which is coded and described in program RPTFM1.PRG. REPORTS.PRG prints a menu which offers the user the following options:

10. Exit and return to START.PRG and display the main menu.
11. Index and PROSPECT database on the cadre rep and prospect name fields, and call RPTFM1.PRG to print the report (Appendix F).
12. Call CONREP1.PRG to provide a report of PROSPECTS not contacted within the last 30, 60, or 90 days (Appendix G). This report can be directed to the

screen or printer and records marked for deletion may be included or omitted from the report.

13. Index the PROSPECT database on high school graduation year and prospect name fields, and call RPTFM1.PRG to print the report (Appendix H).
14. Index the PROSPECT database on region and prospect name fields and call RPTFM1.PRG to print the report (Appendix I).
15. Index the PROSPECT database by high school name and prospect name fields and call RPTFM1.PRG to print the report (Appendix J).
16. Index the PROSPECT database by city and prospect name fields and call RPTFM1.PRG to print the report (Appendix K).
17. Index the PROSPECT database by scholarship types and prospect the name fields and call RPTFM1.PRG to bring the report (Appendix L).
18. Index the PROSPECT database by the source and prospect name fields and call RPTFM1.PRG to print the report (Appendix M).
19. Index the PROSPECT database by the prospect name field and call RPTFM1.PRG to print a report (Appendix N).
20. Call program LABELIT.PRG to print one or more sets of mailing labels (Appendix O).

21. Allows the user to select a specified record from the PROSPECT database and calls INDIV.PRG to print a report (Appendix P).
22. Uses the CONTACT database to create a report of contacts with prospects grouped by contact rep name and indexed by contact rep and prospect name fields (Appendix Q). This report can be directed to either the screen or printer, and deleted records may be included or omitted from the report.

Once an above report is printed or displayed, the user is returned to START.PRG and the main menu is displayed.

RPTFM1.PRG: (Source Code at Appendix U) This program generates a format used by a majority of the predesigned reports. It is called by program REPORTS.PRG and returns to REPORTS.PRG upon completion.

The report is formatted for a 130 character line and is directed to a printer in condense mode. The report is modeled after a similar report in the Cadet Management Database System, and its format and design were dictated by the ROTC department. The report is not designed for screen output.

The user is provided the option of printing or omitting records marked for deletion.

LABELIT.PRG: (Source Code at Appendix V) This program prints mailing labels from records in the PROSPECT database. It is called by REPORTS.PRG and returns to REPORTS.PRG on completion. The program offers the user the choice of generating one of the more sets of labels from records in the PROSPECT database.

FOLLOWUP.PRG: (Source Code at Appendix W) This program is called by START.PRG when option 4 is selected by the user. The program generates a menu and carries out input and editing tasks as directed by the user. The program accesses the FOLLOWUP database and provides the user with the following options:

0. Exit to START.PRG and display the main menu. During the exit process, this program calls NFOL.PRG which scans the FOLLOWUP database and updates the PROSPECT database with the Next Followup Date and the Next Followup Type for each prospect identified in both databases. A notice is displayed on the screen advising the user that the delay in returning to the main menu is due to the update process.
1. Input records into the FOLLOWUP database using a predesigned input screen format (FOLLOW1.SCR).
2. Edit all records in the database associated with a specific prospect name. The user is promoted to

enter the prospect's last name, first name, and middle initial. Once entered, the program uses the LOCATE command to locate all records in the database for the specified prospect. The records are displayed on the screen using the FOLLOW1.SCR format.

CONTACTS.PRG: (Source Code at Appendix X) This program is called by START.PRG when option 5 is selected by the user. Upon completion, it returns to START.PRG and the main menu is displayed.

The program is used to generate a menu and carry out input and editing tasks as directed by the user. The program accesses the CONTACTS database and provides the user with the following options:

0. Exit to the main menu. During the exit process, this program calls a program (LCON.PRG) which scans the CONTACTS database and updates the Last Contact date field in the PROSPECT database with the most recent contact date identified with the prospect in the CONTACT database. A notice is displayed on the screen advising the user of a delay in returning to the main menu due to the update process.
1. Input a record into the CONTACTS database using a predesigned input screen format (CONTACT1.SCR).

2. Edit all records in the database associated with a specific prospect name. When this option is selected, the user is prompted to enter the prospect's last name, first name and middle initial. Once entered, the program uses the LOCATE command to locate all records in the database associated with the specific prospect name. The records are displayed on the screen using the CONTACT1.SCR format. If the record is not located, a message is displayed advising the user and the CONTACTS menu is returned.
3. Edit all records in the CONTACTS database associated with a specific contact representative name. The user is prompted to enter the last name of the contact representative. Records are displayed in the CONTACT1.SCR format for editing. If the record is not found, the screen is cleared, a not found message is displayed, and the user is returned to the CONTACT program menu.
4. Edit all records in the database using predesigned input format, CONTACT1.SCR. Records are displayed alphabetically, by prospect name.

INDIV.PRG: (Source Code at Appendix Y) This program is called by REPORTS.PRG and returns to REPORTS.PRG upon completion. The program formats a report for a specific prospect record in the PROSPECT database.

The program prompts the user to enter the prospect's last name, first name and middle initial. A LOCATE command is then used to search the database and print the results in a specified format. If the record is not found, a message is displayed to the user, and the user is returned to the main menu (START.PRG).

LCON.PRG: (Source Code at Appendix Z) This program is called by CONTACTS.PRG and returns to the main menu (START.PRG) on completion. The program uses data in the CONTACTS database to update a field in the PROSPECT database.

The program searches the CONTACTS database and identifies the latest contact data (C_Date) for each prospect. This date is then entered in the last contact date (LST_CON_DT) field of the respective record in the PROSPECT database.

A message is also printed to the user advising him of the delay in returning to the main menu due to the update process.

NFOL.PRG: (Source Code at Appendix AA) This program is called by FOLLOWUP.PRG and returns to the main menu (START.PRG) on completion. The program uses data in the FOLLOWUP database to update fields in the PROSPECT database.

The program searches the FOLLOWUP database and identifies the next uncompleted followup date (FOL_UP_DAT) for each prospect. This date and its associated followup type (FOL_UP_TYP) are then used to update two fields (NXT_FOL_DT) and (NXT_FOL_TP) in the respective prospect's record in the PROSPECT database.

The program also prints a message to the user advising him of a delay in returning to the main menu due to the update process.

CONREP1.PRG: (Source Code at Appendix BB) This program is called by REPORTS.PRG and returns to REPORTS.PRG upon completion.

The program formats a report based on the last contact date (LST_CON_DT) field in the PROSPECT database.

The program prompts the user to select contacts greater than 30, 60, or 90 days old. The resulting report is grouped by contact representative and can be sent either to the screen or to the printer. The program also gives the user the option of including or omitting records marked for deletion in the report.

Conclusion

The above system was delivered to the KSU Army ROTC Department with complete documentation. The documentation (See Appendix CC) illustrates all the screen formats and describes, in detail, all system operations.

The system was installed on the Zenith PC and demonstrated for the primary departmental users. The users have been satisfied that the PROSPECT system as designed and implemented completely satisfies all user requirements. As of this date, no significant problems nor deficiencies have been reported.

Chapter 5

Conclusions and Future Work

Conclusions

This study illustrated that the KSU Adhoc Design Method produces a schema in Domain Key Normal Form. However, resulting schema may not be the optimum schema for the design. For example, it may contain more than the minimal number of relations and/or more than the minimal number of attribute occurrences. It does, however, satisfy Domain Key criteria that constraints (Functional Dependencies and Multivalued Dependencies) are logical consequences of domains and keys. This finding is significant since it presents a possible option for mechanically transforming lower normal form schemas into Domain Key Normal Form.

The study also resulted in the design and implementation of an automated database system to support the KSU Army ROTC Department's recruiting activities. The design was fully implemented and met all user requirements. It was delivered with full documentation.

Future Work

Future work resulting from this study can be divided into two categories, Theoretical and Implementation. Theoretical work relates to future work regarding design theory and methods. Implementation involves future work

with regard to the actual system implementation. Each category is addressed separately below.

Theoretical

This study showed one example of how the KSU Adhoc Design Method produced a Domain Key Normal Form schema. This one example is by no means a certification that the KSU Method always achieves this same result, especially if embedded MVDs arise. However, the technique of creating separate relations for each MVD appears to satisfy the Domain Key criteria. The proof of this fact remains a subject for future study.

If it is proved that a Domain Key schema always results from a KSU Adhoc Design, then the method should be studied as a possible mechanical means for transforming lower level schemas into Domain Key. No such method currently exists and it is therefore an important area of future work.

Implementation

The PROSPECT system was delivered with eleven preformatted reports. Additional unspecified reports will be required to support future recurring or one-time requirements. One specific report which was not developed was a statistical report required on a periodic basis by the regional headquarters. This report was not designed because

of its constantly changing requirements. Given the capabilities of the dBase III Plus software and the documentation provided with the PROSPECT system, department personnel have the ability to create these reports and other non-specified reports as required.

The PROSPECT system was developed using IBM compatible dBase III Plus software. However, the WANG system which was recently installed in the ROTC department, is not IBM compatible. This compatibility problem was a concern during the design phase, but it was remedied with the acquisition of IBM emulation software for each WANG PC. During system implementation, attempts to invoke the IBM emulation package were not successful. Department personnel were optimistic that future attempts would be successful with the aid of regional headquarters personnel. The capability to run the PROSPECT system on the WANG system greatly affects the availability of the system and as such it is an important aspect of the system design. Every attempt must be made by the department to implement the system on the WANG hardware. This is an important future endeavor.

During the design and implementation of the PROSPECT system, the ROTC department was provided access to an online national database system which contained data on prospective cadets. This system provided a majority of the data required on externally recruited sources, but it did not contain specific data required by the ROTC department to

support its recruiting effort. Therefore, the design of the PROSPECT system continued, and the department decided to support both systems eventhough some duplication of data was required. The duplication of data could be minimized with the development of specialized software to interface between the PROSPECT and national systems. Such development would result in a considerable savings in manhours and it would provide some protection against data inconsistency between the two databases.

All internally developed automated database systems within the Army ROTC department are currently run on the Zenith PC. This IBM compatible computer has two floppy disk drives but does not have a permanent hard disk capability. As a result, significant time is wasted by swapping system and data disks in and out of the disk drives. This situation could be resolved with the acquisition of a 20 or 40 megabyte hard disk for the Zenith System. The acquisition of such a device would greatly enhance the efficiency of operations and it is strongly recommend to the department for action.

The FOLLOWUP database was designed as part of the PROSPECT system to allow department personnel to track follow-up actions for each prospective cadet. With some modification, this database and appropriate programs could serve as a template for a general suspense system. The system could be designed to track future pending actions and

provide reports on a daily or periodic basis of uncompleted pending actions. These functions are similar to those performed by programs in the PROSPECT system using the FOLLOWUP database. Modification of the FOLLOWUP database structure and appropriate programs would not be difficult and it could provide the department with a valuable general purpose management tool to track active suspense actions.

Bibliography

1. Kroenke, David, "Database Processing," Science Research Inc., 1983, Page 21.
2. Unger, Elizabeth A., Lecture Notes, Computer Science 761, Database Management Systems, summer semester, 1987.
3. Fisher, Paul, Lecture Notes, Computer Science 860, Distributed Data Base Systems, fall semester, 1987.
4. Korth, Henry F., Silberschatz, Abraham, "Database System Concepts," McGraw-Hill Book Company, 1986, Page 215.
5. Date, C.J., "An Introduction to Database Systems," Addison-Wesley Publishing Company, 1986, Page 243.
6. Korth, Henry F., Silberschatz, Abraham, "Database System Concepts," McGraw-Hill Book Company, 1986, Page 198.
7. Date, C.J., "An Introduction to Database Systems," Addison-Wesley Publishing Company, 1986, Page 389.
8. Fagin, Ronald, "A Normal Form for Relational Databases That is Based on Domains and Keys," ACM Transaction on Database Systems, Volume 6, Number 3, September 1981, Pages 387-415.
9. Kroenke, David, "Database Processing," Science Research Inc., 1983, Page 299.
10. Kroenke, David, "Database Processing," Science Research Inc., 1983, Page 304.
11. Bernstein, P.A., "Synthesizing Third Normal Form Relations from Functional Dependencies," ACM Transactions on Database Systems, Volume 1, Number 4, December 1976, Pages 277-298.

Appendix A

PROSPECT Structure

Structure for database: C: prospect.dbf

Number of data records: 8

Date of last update: 06/14/88

Field	Field Name	Type	Width	Dec
1	LAST_NAME	Character	20	
2	FIRST_NAME	Character	10	
3	MIDDLE_INITIAL	Character	2	
4	ADMITTED	Character	1	
5	AREA_CODE	Numeric	3	
6	CADET_REP	Character	20	
7	CITY	Character	20	
8	CADRE_REP	Character	10	
9	COLLEGE_CR	Numeric	3	
10	COL_GPA	Numeric	4	2
11	COMMENTS	Character	250	
12	DOB	Date	8	
13	ENTRY_DATE	Date	8	
14	DISPOSITION	Character	2	
15	DISPOSITION_DATE	Date	8	
16	NXT_FOL_DT	Date	8	
17	NXT_FOL_TP	Character	20	
18	LST_CON_DT	Date	8	
19	HI_SCHOOL	Character	30	
20	SCHOOL_DIS	Numeric	3	
21	HS_GRAD_YR	Numeric	2	
22	IN_GOLD_QS	Character	1	
23	COL_MAJOR	Character	20	
24	MIL_EXPER	Character	4	
25	PHONE_NUMBER	Numeric	8	
26	REFER_BY	Character	20	
27	REGION	Numeric	1	
28	SCHOLARSHIP_AP	Character	1	
29	SEX	Character	1	
30	SOURCE	Character	2	
31	SSN	Numeric	9	
32	STATE	Character	2	
33	STREET	Character	20	
34	TITLE	Character	4	
35	ZIP	Numeric	9	
36	RACE	Character	1	
37	INFLUENCER	Character	20	
** Total **			564	

Appendix B

FOLLOWUP Structure

Structure for database: C: followup.dbf
Number of data records: 12
Date of last update: 06/14/88

Field	Field Name	Type	Width	Dec
1	P_L_NAME	Character	20	
2	P_F_NAME	Character	10	
3	P_M_INITIAL	Character	2	
4	FOL_UP_DATE	Date	8	
5	FOL_UP_TYPE	Character	20	
6	COMPLETED	Character	1	
**	Total	**	62	

Appendix C

CONTACT Structure

Structure for database: C: contact.dbf
Number of data records: 11
Date of last update: 06/14/88

Field	Field Name	Type	Width	Dec
1	C_DATE	Date	8	
2	C_REP	Character	10	
3	C_METHOD	Character	1	
4	C_REMARKS	Character	30	
5	P_L_NAME	Character	20	
6	P_F_NAME	Character	10	
7	P_M_INITIAL	Character	2	
**	Total	**	82	

Appendix D

Data Dictionary

Data Name: ADMITTED

Aliases:

Data Type: character

Format: one byte

Range: either "A", "P", "N", or blank

Responsibility: Recruiting officer

Frequency: as determined

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database. It indicates the status of the prospect's application to the college of his choice. "A" = Admitted, "P" = Applied, "N" = Not Applied. This field may also be blank.

Data Name: AREA_CODE

Aliases: Area Code

Data Type: numeric

Format: ###

Range: 111 - 999 or may be blank

Responsibility: Recruiting officer

Frequency: as required

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database. It contains the prospect's telephone area code.

Data Name: CADET_REP

Aliases: Cadet Representative

Data Type: character string

Format: 10 bytes, all capital letters

Range: unrestricted

Responsibility: Recruitment officer

Frequency: as assigned

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database. It identifies the name of the Cadet Representative assigned to the prospect. This field may be blank.

Data Name: CADRE_REP

Aliases: Cadre Representative

Data Type: character string

Format: ten bytes

Range: all capital letters

Responsibility: Recruitment officer

Frequency: as determined

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database. It identifies the name of the ROTC cadre officer assigned to the prospect. One Rep will be assigned to many prospects, but each prospect has only one cadre representative.

Data Name: CITY

Aliases: Town

Data Type: character string

Format: 20 bytes

Range: no restrictions

Responsibility: Recruiting officer

Frequency: as required

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database. It identifies the city where the prospect lives.

Data Name: COLLEGE_CR

Aliases: College Credits

Data Type: numeric

Format: ###

Range: 000 - 999 or may be blank

Responsibility: Recruiting officer

Frequency: as required

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database. It identifies the number of college credits the prospect has accumulated.

Data Name: COL_GPA

Aliases: College Grade Point Average

Data Type: numeric

Format: #.##

Range: 0.00 - 4.00 or may be blank

Responsibility: Recruiting officer

Frequency: as required

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database. It contains the college GPA earned by the prospect if he has completed college courses.

Data Name: COL_MAJOR

Aliases: College Major

Data Type: character string

Format: 20 bytes

Range: no restrictions

Responsibility: Recruiting officer

Frequency: as required

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database. If completed it identifies the prospects planned or present major field of study.

Data Name: COMMENTS

Aliases:

Data Type: character string

Format: 250 bytes

Range: no restrictions

Responsibility: Cadre Representative

Frequency: as required

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database. It is used for general comments about each prospect.

Data Name: COMPLETED

Aliases: Follow-up Completed

Data Type: one byte

Format: character

Range: "Y" = Yes, "N" = No

Responsibility: Recruitment officer

Frequency: As determined

Dependencies: Multi-determined by the Prospect's Name

Comments: This field is in the FOLLOWUP database. It is used to record the completion of a follow-up action. "Y" = Yes, "N" = Not Completed

Data Name: C_DATE

Aliases: Contact date

Data Type: date

Format: mm/dd/yy

Range:

Responsibility: Contacting representative

Frequency: after each contact with the prospect

Dependencies: With Contact Method and the Prospect's name it determines Contact Rep and Contact Remarks. With Contact Rep, it Multidetermines names of Prospects. This field is also used to automatically update the Last Contact Date field in the PROSPECT database.

Comments: The field is in the CONTACT database. It is used to automatically update the Last Contact Date field in the PROSPECT database (See description for LST_CON_DT). This field is used to indicate the date on which a contact was made with a prospect.

Data Name: C_METHOD

Aliases: Contact Method

Data Type: character

Format: one byte

Range: one of the following: "P", "T", "M", or "N"

Responsibility: Contacting officer

Frequency: after each contact with the prospect

Dependencies: With Contact Date, Contact Method and Prospects Name, it will determine Contact Rep and Contact Rmks

Comments: This field is in the CONTACT database. "P" = In Person, "T" = Telephone, "M" = Mail, and "N" = Not Contacted, Blank is Not Allowed.

Data Name: C_REMARKS

Aliases: Contact Remarks

Data Type: character string

Format: 30 bytes

Range: no restriction

Responsibility: Contacting officer

Frequency: after each contact with the prospect

Dependencies: determined by Prospect Name, Contact Date, Contact Representative, and Contact Type

Comments: This field is in the CONTACTS database. It contains remarks by a Contacting officer regarding a particular contact with a prospect.

Data Name: C_REP

Aliases: Contact Representative

Data Type: character string

Format: 10 bytes

Range: all capital letters

Responsibility: Contacting officer

Frequency: after each contact with the prospect

Dependencies: With Contact Date it Multidetermines the Prospect. It also is determined by Contact Date, Contact Method, and the Prospect's Name.

Comments: This field is in the CONTACT database. It identifies the individual within the ROTC department who contacts a prospect.

Data Name: DISPOSITN

Aliases: Disposition

Data Type: character string

Format: two bytes

Range: one of the following: "EN", "NI", "NQ" or "OO".

Responsibility: Recruitment Officer

Frequency: as determined

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database. It indicates the prospect's final disposition regarding enrollment in the ROTC program. "EN" = Enrolled, "NI" = Not Interested, "IN" = Interested, "NQ" = Not Qualified, "OO" = Other.

Data Name: DISPO_DATE

Aliases: Disposition Date

Data Type: date

Format: mm/dd/yy

Range:

Responsibility: Recruitment officer

Frequency: as determined

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database. It indicates the date that a final disposition of the prospect was determined.

Data Name: DOB

Aliases: Date of Birth

Data Type: date

Format: mm/dd/yy

Range:

Responsibility: Recruiting officer

Frequency: as determined

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database. It contains the prospect's date of birth. It can be used along with the system date to determine the prospects age.

Data Name: ENTRY_DATE

Aliases: Date, Date of Referral

Data Type: date

Format: mm/dd/yy

Range:

Responsibility: Recruiting officer

Frequency: As required

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database. It is the date the individual was identified as a prospective cadet and entered into the system.

Data Name: FIRST_NAME

Aliases: none

Data Type: character string

Format: 10 bytes or less

Range: none

Responsibility: Recruiting officer

Frequency: as required

Dependencies: along with Last Name and MI it forms the primary key to a prospect's record.

Comments: This field is in the PROSPECT database. It is associated with the persons last name and middle initial to make up the primary key to a prospect's record.

Data Name: FOL_UP_DAT

Aliases: Follow-up date

Data Type: date

Format: mm/dd/yy

Range:

Responsibility: Recruitment officer

Frequency: as determined

Dependencies: determined by Prospect's Name

Comments: This field is in the FOLLOWUP database. It is used to automatically update the Next Followup Date field in the PROSPECT database.

Data Name: FOL_UP_TYP

Aliases: Follow-up Type

Data Type: character string

Format: 20 bytes

Range: no restrictions

Responsibility: Recruitment officer

Frequency: as determined

Dependencies: determined by Prospect's Name and Followup Date.

Comments: This field is in the FOLLOWUP database. It is used to automatically update the Next Followup Type field in the PROSPECT database.

Data Name: HI_SCHOOL

Aliases: High School

Data Type: character string

Format: 30 bytes

Range: no restrictions

Responsibility: Recruiting officer

Frequency: as required

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database. It contains the name of the high school attended by the prospect.

Data Name: HS_GRAD_YR

Aliases: High School Graduation Year

Data Type: numeric

Format: yy

Range: 85 - 99

Responsibility: Recruiting officer

Frequency: as required

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database. It contains the high school graduation year of the prospect.

Data Name: INFLUENCER

Aliases:

Data Type: character string

Format: 20 bytes

Range: unrestricted

Responsibility: Recruiting officer

Frequency: as determined

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database. It contains the name of a person who can influence the prospect's decision to join ROTC.

Data Name: IN_GOLD_QS

Aliases: In Gold Quest

Data Type: character

Format: one byte

Range: either "Y" or "N"

Responsibility: Recruiting officer

Frequency: as determined

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database. "Y" indicates the prospect is in the National Gold Quest System. "N" indicates he is not in the National Gold Quest System. The Gold Quest System was formerly called the National Leads System.

Data Name: LAST_NAME

Aliases: none

Data Type: Character

Format: 20 bytes

Range: NA

Responsibility: Recruiting officer

Frequency: as required

Dependencies: along with First Name and MI it forms the primary key to a prospect's record.

Comments: This field is in the PROSPECT database. It contains the Last Name of the prospective cadet and along with First Name and Middle Initial it forms the key to each prospect's record.

Data Name: LST_CON_DT

Aliases: Last Contact Date

Data Type: date

Format: mm/dd/yy

Range:

Responsibility: Program generated

Frequency: automatically updated when a Contact Date is entered in the CONTACT dbf

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database. It contains the date the prospect was last contacted by a Contact Rep. The field is automatically updated whenever a contact is added or edited in the CONTACT database.

Data Name: MIDDLE_INI

Aliases: Middle Initial

Data Type: character

Format: 2 bytes

Range: no restriction

Responsibility: Recruiting Officer

Frequency: as required

Dependencies: along with First Name and Last Name it forms the primary key to a prospect's record.

Comments: This field is in the PROSPECT database. It is included with the Last Name, and First Name fields to form the key to the prospect's record. The second byte of the field is used to distinguish between two prospects with the same name.

Data Name: MIL_EXPER

Aliases: Military Experience

Data Type: character string

Format: four bytes

Range: one of the following: "AA-C", "AA-P", "AR-C", "AR-P", "NG-C", "NG-P", "USNC", "USNP", "MC-P", "MC-C", "AF-C", "AF-P", "OTHR", or blanks

Responsibility: Recruiting officer

Frequency: as required

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database. It is completed if the prospect has had prior military training or experience. First two/three letters indicate branch of service: AA = Active Army, AR = Reserves, NG = National Guard, USN = Navy, AF = AirForce, MC = Marine Corps, OTHR = Other. The last character is either a C or P. C = Current, and is used if the soldier is currently on active duty with the service; P = Prior and indicates prior service with that Branch. This field may be blank.

Data Name: NXT_FOL_DT

Aliases: Next Follow-up Date

Data Type: date

Format: mm/dd/yy

Range:

Responsibility: Program generated

Frequency: Entered and updated automatically whenever a followup date is entered or updated in the FOLLOWUP database.

Dependencies: determined by and entry or update to the Follow-Up-Date in the FOLLOWUP database.

Comments: This field in the PROSPECT database is entered or updated automatically from and entry or update to the FOL_UP_DAT field in the FOLLOWUP database. It is program generated.

Data Name: NXT_FOL_TP

Aliases: Next Follow-up Type

Data Type: character string

Format: 20 bytes

Range: no restrictions

Responsibility: Program generated

Frequency: Entered or updated automatically from the FOLLOWUP database each time a followup activity is entered or updated.

Dependencies: determined by Followup type in the FOLLOWUP database.

Comments: This field is in the PROSPECT database. It provides the next type of follow-up action required for a prospect. It is automatically entered and updated each time an entry is input or updated in the FOLLOWUP database. It contains the type of the most recent non-completed follow-up action.

Data Name: PHONE_NUM

Aliases: Phone Number

Data Type: numeric

Format: ###-####

Range: 111-1111 to 999-9999 or may be blank

Responsibility: Recruiting officer

Frequency: as required

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database. It contains the prospect's home telephone number without area code.

Data Name: P_F_NAME

Aliases: First Name

Data Type: character

Format: 10 bytes

Range: unrestricted

Responsibility: Recruitment or Contacting Officer

Frequency: as required

Dependencies: along with P_L_NAME and P_M_INI determines contacts and followups in the respective databases.

Comments: This field is found in both the FOLLOWUP and CONTACT databases. It contains the first name of the prospect on which a followup or contact action has occurred. This name should correspond to the field FIRST_NAME in the PROSPECT database.

Data Name: P_L_NAME

Aliases: Last Name

Data Type: character

Format: 20 bytes

Range: unrestricted

Responsibility: Recruitment or Contacting Officer

Frequency: as required

Dependencies: along with P_F_Name and P_M_INI determines contacts and followups in the respective databases.

Comments: This field is found in both the FOLLOWUP and CONTACT databases. It contains the last name of the prospect on which a followup or contact action has occurred. This name should correspond to the field LAST_NAME in the PROSPECT database.

Data Name: P_M_INI

Aliases: Middle Name

Data Type: character

Format: 2 bytes

Range: unrestricted

Responsibility: Recruitment or Contacting Officer

Frequency: as required

Dependencies: along with P_F_NAME and P_L_NAME determines contacts and followups in the respective databases.

Comments: This field is found in both the FOLLOWUP and CONTACT databases. It contains the middle name of the prospect on which a followup or contact action has occurred. This name should correspond to the field MIDDLE_INI in the PROSPECT database.

Data Name: RACE

Aliases:

Data Type: character

Format: one byte

Range: Uses standard race code: C, N, R, M, X, or Z. See comments

Responsibility: Recruiting officer

Frequency: as determined

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database. The following codes are accepted C = White (Caucasoid), N = Black (Negro/African), R = Red (American Indian), M = Yellows (Asian/Mongoloid), X = Other, Z = Unknown. These are the codes used on the TRADOC form 471.

Data Name: REFER_BY

Aliases: Referred by/Referring agency

Data Type: character string

Format: 20 bytes

Range: no restrictions

Responsibility: Recruiting officer

Frequency: as determined

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database. It indicates the name of the agency or person who referred the prospect to the KSU ROTC Department.

Data Name: REGION

Aliases: Region

Data Type: numeric

Format: #

Range: 0 - 9 or blank

Responsibility: Recruitment officer

Frequency: as determined

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database. It is used to indicate the KSU Admissions recruiting region code.

Data Name: SCHOLAR_AP

Aliases: Scholarship Application

Data Type: character

Format: one byte .

Range: One of the following "4", "3", "2", "A", "N", "O"

Responsibility: Cadre Representative

Frequency: as determined

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database. It is used to indicate whether a scholarship application has been provided to the prospect, and if so, what type. 4 = four years, 3 = three years, 2 = two year, A = active army, N = national guard, O = other.

Data Name: SCHOOL_DIS

Aliases: School District

Data Type: numeric

Format: ###

Range: 000 - 999 or blank

Responsibility: Recruiting officer

Frequency: as determined

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database. It contains the school district number where the prospect attends high school. This field may be blank.

Data Name: SEX

Aliases:

Data Type: character

Format: one byte

Range: either "M" for Male or "F" for Female

Responsibility: Recruiting officer

Frequency: as determined

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database.

Data Name: SOURCE

Aliases: none

Data Type: character

Format: two bytes

Range: one of the following "IN" for Internal, "GQ" for Gold Quest, "GG" for Green to Gold, "UR" for US Army Recruiting Command, "GO" for Goldminer, and "OO" for Other.

Responsibility: Recruitment officer

Frequency: as determined

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database. It identifies the agency which referred the prospect to the KSU ROTC department.

Data Name: SSN

Aliases: Social Security Account Number

Data Type: Numeric

Format: #####

Range: May be blank

Responsibility: Recruiting officer

Frequency: as determined

Dependencies: determined by Prospect's Name and when not blank determines the Prospect's Name.

Comments: This field is in the PROSPECT database. It is not a key field since data to fill this field is not always available, and it may therefore be blank.

Data Name: STATE

Aliases:

Data Type: character string

Format: two bytes

Range: unrestricted

Responsibility: Recruiting officer

Frequency: as required

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database. It contains the two character code of the State where the prospect resides.

Data Name: STREET

Aliases: Street address

Data Type: character string

Format: 20 bytes

Range: no restrictions

Responsibility: Recruiting officer

Frequency: as required

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database. It identifies the prospect's street address.

Data Name: TITLE

Aliases: none

Data Type: character string

Format: four bytes

Range: one of the following "Mr" "Mrs" "Miss" "Ms"

Responsibility: Recruiting officer

Frequency: as required

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database. It is used primarily for printing labels and addressing correspondence.

Data Name: ZIP

Aliases: Zip code

Data Type: numeric

Format: #####-####

Range: 00000-0000 to 99999-9999 or may be blank

Responsibility: Recruiting officer

Frequency: as required

Dependencies: determined by Prospect's Name

Comments: This field is in the PROSPECT database. It contains the prospect's zip code.

Appendix E

State Codes

Two Character State Codes

AL - Alabama	MT - Montana
AK - Alaska	NE - Nebraska
AZ - Arizona	NV - Nevada
AR - Arkansas	NH - New Hampshire
CA - California	NJ - New Jersey
CO - Colorado	NM - New Mexico
CT - Connecticut	NY - New York
DE - Delaware	NC - North Carolina
DC - District of Columbia	ND - North Dakota
FL - Florida	OH - Ohio
GA - Georgia	OK - Oklahoma
GU - Guam	OR - Oregon
HI - Hawaii	PA - Pennsylvania
ID - Idaho	PR - Puerto Rico
IL - Illinois	RI - Rhode Island
IN - Indiana	SC - South Carolina
IA - Iowa	SD - South Dakota
KS - Kansas	TN - Tennessee
KY - Kentucky	TX - Texas
LA - Louisiana	UT - Utah
ME - Maine	VT - Vermont
MD - Maryland	VI - Virgin Islands
MA - Massachusetts	VA - Virginia
MI - Michigan	WA - Washington
MN - Minnesota	WV - West Virginia
MS - Mississippi	WI - Wisconsin
MO - Missouri	WY - Wyoming

Appendix F

Cadre Representative Report

Page No. 1	ROTC Prospect--CADRE Representative Report				06/15/88		
Name of Prospect	Date	Military No	Source	Cadre Rep	Contact	Follow	Follow Type
Address	Gold Dst	Scholar	School				
Phone Number	Region	Grad Yr	Comments				
Baker, Jane B 22 5th Street Los Alamo, AZ 987656545 (888) 345-9876	05/14/88	AA-P 4 85	IN Los Alamo	06/06/88	08/03/88	Initial Let.
Doe, John P 25 Maple Street Manhattan, KS 66502 (913) 555-1213	06/09/88 4 89	IN Manhattan	BATES	06/09/88	07/20/88	Initial Let. Excellent Prospect. Good leadership and academic record.
Johnson, James P 45 8th Street San Pedro, CA 99859 (945) 234-9898	06/06/88	USNP 2 85	UR San Pedro	BATES	06/07/88	08/08/88	Scholar. App. Good potential. Prior service. Gold military record.
Penn, Harry H 45 West Street Manhattan, KS 66502 322-3232	06/07/88 4 89	IN Manhattan	BATES	06/07/88	/ /
Smith, Connie M 898 South Street Jacksonville, FL 44440000 776-9999	01/01/88	AF-P 4 88	GO Jacksonville	BATES	06/03/88	09/15/88	Visit Good potential. Prior Service. Good military record.

Appendix G

Last Contact Report

Page No. 1
06/15/88

Contacts Overs 1 month old

Grouped
by
Cadre Representative

PROSPECTS NAME	LAST CONTACT DATE	TELEPHONE	ADDRESS
**CADRE REP BROWN Leonard Bobby R	04/04/88(112)	5327841	16 Yates St. Harv. FL

Appendix H

High School Graduation Year Report

Page No. 1	ROTC Prospect--High School Graduation Year Report				06/15/88
Name of Prospect Address Phone Number	Date Gold est Region	Military Scholar grad Yr	Source School Comments	Cadre Rep Contact Follow	Follow Type
Baker, Jane B 22 5th Street Los Alamo, AZ 987656545 (888) 345-9876	05/14/88 No ...	AA-P 4 85	IN Los Alamo 06/06/88 08/03/88	Initial Let.
Johnson, James P 45 8th Street San Pedro, CA 89899 (945) 234-9898	06/06/88 No ...	USNP 2 85	UR San Pedro	BATES 06/07/88 08/08/88	Scholar. App. Good potential. Prior service. Gold military record.
Smith, Connie M 898 South Street Jacksonville, FL444440000 776-9999	01/01/88 No ...	AF-P 4 88	CO Jacksonville	BATES 06/03/88 09/15/88	Visit Good potential. Prior service. Good military record.
Albertson, James I 3 North Street Junction City, KS 88888 (212) 555-8917	06/01/88 No 4	AD-A a 89	GQ Junction City	BROWN 06/06/88 07/07/88	Scholar. App.
Doe, John P 25 Maple Street Manhattan, KS 66502 (913) 555-1213	06/09/88 No 4 4 89	IN Manhattan	BATES 06/09/88 07/20/88	Initial Let. Excellent Prospect. Good leadership and academic record.

Appendix I

Region Report

Page No. 1

ROTC Prospect--Region Report

Name of Prospect Address Phone Number	Date Gold Est Region	Military Scholar Grad Yr	Source School Comments	Cadre Rep Contact	Follow Contact	Follow Type Initial Let.
Baker, Jane B 22 5th Street Los Alamo, AZ 987656545 (888) 345-9876	05/14/88 No ...	AA-P 4 85	IN Los Alamo	06/06/88 08/03/88	Initial Let.
Johnson, James P 45 8th Street San Pedro, CA 89899 (945) 234-9898	06/06/88 No ...	USNP 2 85	UR San Pedro Good potential. Prior service. Gold military record.	BATES	06/07/88 08/08/88	Scholar. App.
Jones, John P 788 4th Street Huntsville, AL 999999999 (913) 555-5555	06/01/88 No 4 89	IN Huntsville High School	BROWN	06/02/88 09/13/88	Letter
Leonard, Bobby R 16 Yates St Harvard, FL 88888 (112) 532-7841	06/01/88 No 89	GQ Harvard	BROWN	04/04/88 / /
Smith, Connie M 898 South Street Jacksonville, FL 44440000 776-9999	01/01/88 No ...	AF-P 4 88	GO Jacksonville Good potential. Prior Service. Good military record.	BATES	06/03/88 09/15/88	Visit

Appendix J

High School Report

Page No. 1 ROTC Prospect--High School Report 06/15/88

Name of Prospect Address Phone Number	Date Gold St Region	Military Source Scholar Grad Yr	Cadre Rep School Comments	Contact Follow	Follow Type
Leonard, Bobby R 16 Yates St Harvard, FL 88888 (112) 532-7841	06/01/88 No	GO Harvard	BROWN 04/04/88 /
Jones, John P 788 4th Street Huntsville, AL 999999999 (913) 595-5555	06/01/88 No	IN Huntsville High School	BROWN 06/02/88 05/13/88	Letter
Smith, Connie M 898 South Street Jacksonville, FL 444440000 776-9999	01/01/88 No ...	AF-P 4 88	GO Jacksonville Good potential. Prior Service.	BATES 06/03/88 09/15/88	Visit Good military record.
Albertson, James I 3 North Street Junction City, KS 88888 4 (212) 595-8917	06/01/88 No ...	AD-A 8 89	GO Junction City	BROWN 06/06/88 07/07/88	Scholar. App.
Baker, Jane B 22 5th Street Los Alamo, AZ 987656545 (888) 345-9876	05/14/88 No ...	AA-P 4 85	IN Los Alamo 06/06/88 08/03/88	Initial Let.

Appendix K

City Report

Page No. 1

Name of Prospect		Date		Military Source		Cadre Rep		Contact		Follow		Type	
Address		No		No		School		Follow		Follow		Follow	
Phone Number		Region		Grad Yr		Comments		Contact		Contact		Type	
#####													
Leonard, Bobby R		06/01/88			GQ		BROWN		04/04/88		/ /	
16 Yates St		No		89		Harvard						
Harvard, FL 88888		...											
(112) 532-7841													
#####													
Jones, John P		06/01/88			IN		BROWN		06/02/88		09/13/88	
788 4th Street		No		4		Huntsville		High School				Letter	
Huntsville, AL 999999999		...		89									
(913) 555-5555													
#####													
Smith, Connie M		01/01/88		AF-P		GO		BATES		06/03/88		09/15/88	
898 South Street		No		4		Jacksonville						Visit	
Jacksonville, FL 44440000		...		88								Good potential. Prior Service. Good military record.	
..... 776-9999													
#####													
Albertson, James I		06/01/88		AD-A		CQ		BROWN		06/06/88		07/07/88	
3 North Street		No		a		Junction City						Scholar. App.	
Junction City, KS 88888 4		...		89									
(212) 555-8917													
#####													
Baker, Jane B		05/14/88		AA-P		IN			06/06/88		08/03/88	
22 5th Street		No		4		Los Alamo						Initial Let.	
Los Alamo, AZ 987656545		...		85									
(888) 345-9876													

Appendix L

Scholarship Type Report

Page No. 1

ROTC Prospect--Scholarship Type Report

06/15/88

Name of Prospect	Date	Military	Source	Cadre Rep	Contact	Follow	Follow Type
Address	Gold Est	Scholar	School				
Phone Number	Region	Grad Yr	Comments				

Leonard, Bobby R	06/01/88	GQ	BROWN	04/04/88	/
16 Yates St	No	89	Harvard				
Harvard, FL 88888	...						
(112) 532-7841							

Johnson, James P	06/06/88	USNP	UR	BATES	06/07/88	08/08/88	Scholar. App.
45 8th Street	No	2	San Pedro				
San Pedro, CA 90899	...	85	Good Potential. Prior service. Gold military record.				
(945) 234-9898							

Baker, Jane B	05/14/88	AA-P	IN	06/06/88	08/03/88	Initial Let.
22 5th Street	No	4	Los Alamo				
Los Alamo, AZ 987656545	...	85					
(888) 343-9876							

Doe, John P	06/09/88	IN	BATES	06/09/88	07/20/88	Initial Let.
25 Maple Street	No	4	Manhattan				
Manhattan, KS 66502	4	85	Excellent Prospect. Good leadership and academic record.				
(913) 555-1213							

Jones, John P	06/01/88	IN	BROWN	06/02/88	09/13/88	Letter
788 4th Street	No	4	Huntsville	High School			
Huntsville, AL 999999999	...	89					
(913) 555-5555							

Appendix M

Source Report

Page No. 1

Name of Prospect		Date	Military	Source	Cadre Rep	Contact	Follow	06/15/88
Address		No	Scholar	School			Type	
Phone Number		Region	Grad Yr	Comments				
#####		#####	#####	#####	#####	#####	#####	#####
Smith, Connie M		01/01/88	AF-P	GO	BATES	06/03/88	09/15/88	Visit
898 South Street		No	4	Jacksonville				
Jacksonville, FL 4440000		...	88	Good potential. Prior Service. Good military record.				
..... 776-9999								
#####		#####	#####	#####	#####	#####	#####	#####
Albertson, James I		06/01/88	AD-A	GQ	BROWN	06/06/88	07/07/88	Scholar. App.
3 North Street		No	a	Junction City				
Junction City, KS		88888 4	89					
(212) 555-8917								
#####		#####	#####	#####	#####	#####	#####	#####
Leonard, Bobby R		06/01/88	GQ	BROWN	04/04/88	/ /
16 Yates St		No		Harvard				
Harvard, FL 88888		...	89					
(112) 532-7841								
#####		#####	#####	#####	#####	#####	#####	#####
Baker, Jane B		05/14/88	AA-P	IN	06/06/88	08/03/88	Initial Let.
22 5th Street		No	4	Los Alamo				
Los Alamo, AZ 987656545		...	85					
(888) 345-9876								
#####		#####	#####	#####	#####	#####	#####	#####
Doe, John P		06/09/88	IN	BATES	06/09/88	07/20/88	Initial Let.
25 Maple Street		No	4	Manhattan				
Manhattan, KS 66502		4	89	Excellent Prospect. Good leadership and academic record.				
(913) 555-1213								

Appendix N

Alphabetical Report

Page No. 1	ROTC Prospect--Alphabetical Report		06/15/88
Name of Prospect	Date	Military Source	Follow Type
Address	Gold Est	Scholar	Follow
Phone Number	Region	Grad Yr	Contact
		Comments	
Albertson, James I 3 North Street Junction City, KS 88888 4 (212) 555-8917	06/01/88 No 4	AD-A a 89 GQ Junction City	BROWN 06/06/88 07/07/88 Scholar. App.
Baker, Jane B 22 5th Street Los Alamo, AZ 987656545 (888) 345-9876	05/14/88 No ...	AA-P 4 85 IN Los Alamo 06/06/88 08/03/88 Initial Let.
Doe, John P 23 Maple Street Manhattan, KS 66502 (913) 555-1213	06/09/88 No 4 4 85 IN Manhattan	BATES 06/09/88 07/20/88 Initial Let. Excellent Prospect.Good leadership and academic record.
Johnson, James P 45 8th Street San Pedro, CA 89899 (945) 234-9898	06/06/88 No ...	USNP 2 85 UR San Pedro	BATES 06/07/88 08/08/88 Scholar. App. Good potential. Prior service. Gold military record.
Jones, John P 788 4th Street Huntsville, AL 999999999 (913) 555-3555	06/01/88 No 4 89 IN Huntsville High School	BROWN 06/02/88 09/13/88 Letter

Appendix O

Label Report

Mr John P Jones
788 4th Street
Huntsville AL 999999999

Ms Connie M Smith
898 South Street
Jacksonville FL 444440000

Mr James I Albertson
3 North Street
Junction City KS 88888

Mr Bobby R Leonard
16 Yates St
Harvard FL 88888

Mr James P Johnson
45 8th Street
San Pedro CA 89899

Mr Harry H Penn
45 West Street
Manhattan KS 66502

Mr John P Doe
25 Maple Street
Manhattan KS 66502

Ms Jane B Baker
22 5th Street
Los Alamo AZ 987656545

Appendix P

Individual Report

ARMY ROTC PROSPECT

ENTRY_DATE: 06/01/88

TITLE: Mr LAST_NAME: Jones FIRST NAME: John MI: P

SSN: 234567890 SEX: M RACE: C DOB: 01/01/65 AREA CODE:913
PHONE: 5555555

STREET: 788 4th Street CITY: Huntsville STATE: AL
ZIP: 999999999

HIGH_SCHOOL: Huntsville High School HS_GRAD_YR: 89

SCHOOL_DISTRICT: 0 KSU_RECRUITING_REGION: 0

COLLEGE_CR: 0 COLLEGE_GPA: 0.00 COLLEGE_MAJOR: Engineering

MILITARY_EXPR: SCHOLARSHIP_APPLICATION: 4 ADMITTED: y

SOURCE: IN REFERRED_BY: INFLUENCER:

CADRE_REP: BROWN CADET_REP: LAST_CONTACT_DATE: 06/02/88

NEXT_FOLLOW_UP_DATE: 09/13/88 NEXT_FOLLOW_UP_TYPE: Letter

IN_GOLD_QUEST: DISPOSITION: DISPOSITION_DATE: / /

COMMENTS:

Appendix Q

Contact Representative Report

Page No. 1

06/15/88

Report of Contacts
by
Contact Representative

PROSPECTS NAME	CONTACT DATE	CONTACT METHOD	CONTACT REMARKS
** CONTACT REP BATES			
Doe John P	06/09/88	M	Initial Information Letter
Johnson James P	06/07/88	T	Initial contact
Jones John P	12/13/87	P	High School Visit
Penn Harry H	06/05/88	T	Introductory phone call
Penn Harry H	06/07/88	M	Sent Information Letter
Smith Connie M	03/23/88	P	Great Prospect
Smith Connie M	01/01/88	L	Initial Letter
Smith Connie M	06/03/88	T	Followup phone call
** CONTACT REP BROWN			
Albertson James I	06/06/88	T	Just keeping in touch
Jones John P	06/02/88	T	Initial contact
** CONTACT REP SMITH			
Baker Jane B	06/06/88	M	Scholarship information

Appendix R: START.PRG Code

```
*****
*
*
*           START.PRG
*           by
*           Richard B. Thompson
*           June 8, 1988
*
*****
*
*           This is the starting program for access to
*           the PROSPECT, CONTACT, and FOLLOWUP databases. It
*           prints a menu which offers the user one of seven
*           choices (0 to 7).
*
*           0 -> Exits the program to the dBase Dot
*
*           1 -> Allows the user to input new records. It uses the
*           customized input form PROSPIN1 and appends the
*           new records to the PROSPECT database.
*
*           2 -> Executes program EDITER.PRG, which allows the
*           user to edit records in the PROSPECT database.
*
*           3 -> Executes program REPORTS.PRG, which provides the
*           user with a menu of predesigned reports.
*
*           4 -> Executes program FOLLOWUP.PRG which allows the
*           user to input or edit data in the FOLLOWUP
*           database.
*
*           5 -> Executes program CONTACTS.PRG which allows the
*           user to input or edit data in the CONTACT
*           database.
*
*           6 -> Exits the program to operating system " A> ".
*
*****
*
*
* CLEAR ALL
* CLOSE ALL
* DO WHILE .T.
* SET DELETED OFF
* SET TALK OFF
* SET HEADING OFF
* SET SAFETY OFF
* SET BELL OFF
* SET DELIMITER OFF
* SET STATUS OFF
* SET CONFIRM OFF
*
*
```

*


```

CLEAR
* Print the Menu on the screen
@ 1, 0 SAY "=====
@ 1,40 SAY "=====
@ 2, 0 SAY "||"
@ 2, 6 SAY "                K S U R O T C P R O S P E C T S"
@ 2,78 SAY "||"
@ 3, 0 SAY "=====
@ 3,40 SAY "=====
@ 4, 0 SAY "||"
@ 4,78 SAY "||"
@ 5, 0 SAY "||"
@ 5,78 SAY "||"
@ 6, 0 SAY "||"
@ 6,78 SAY "||"
@ 7, 0 SAY "||"
@ 7,78 SAY "||"
@ 8, 0 SAY "||"
@ 8,78 SAY "||"
@ 9, 0 SAY "||"
@ 9,78 SAY "||"
@ 10, 0 SAY "||"
@ 10,78 SAY "||"
@ 11, 0 SAY "||"
@ 11,78 SAY "||"
@ 12, 0 SAY "||"
@ 12,78 SAY "||"
@ 13, 0 SAY "||"
@ 13,78 SAY "||"
@ 14, 0 SAY "||"
@ 14,78 SAY "||"
@ 15, 0 SAY "||"
@ 15,78 SAY "||"
@ 16, 0 SAY "||"
@ 16,78 SAY "||"
@ 17, 0 SAY "=====
@ 17,40 SAY "=====
@ 5,20 SAY " 0. EXIT to dBase dot prompt"
@ 7,20 SAY " 1. INPUT new Prospect into the database"
@ 8,20 SAY " 2. EDIT PROSPECTS database"
@ 9,20 SAY " 3. PRINT a REPORT"
@ 10,20 SAY " 4. Enter/Edit a FOLLOW-UP action on a Prospect"
@ 11,20 SAY " 5. Enter/Edit a CONTACT action with a Prospect"
@ 13,20 SAY " 6. Leave dBase, EXIT to A>"
STORE 10 TO selectnum
DO WHILE selectnum < 0 .OR. selectnum > 6
  STORE " " TO select
  @ 17,33 SAY " select : : "
  @ 17,42 GET select PICTURE "#"
  READ
  STORE VAL(select) TO selectnum
ENDDO
*
```

```
CLEAR
*
DO CASE
  CASE selectnum= 0
    USE
    CLEAR ALL
    RETURN
  CASE selectnum= 1
    * Input new Prospect
    USE Prospect.dbf
    SET FORMAT TO Prospini
    APPEND
    USE
  CASE selectnum= 2
    * Edit database
    DO EDITER.PRG
  CASE selectnum= 3
    * Print a Report from the database
    DO reports.prg
  CASE selectnum= 4
    * Enter/Edit a follow-up action
    DO followup.prg
  CASE selectnum= 5
    * Enter/Edit a contact action
    DO contacts.prg
  CASE selectnum= 6
    * Leave dBase and exit to A>
    SET DELIMITER ON
    SET BELL ON
    SET TALK ON
    SET HEADING ON
    SET SAFETY ON
    CLEAR ALL
    QUIT
ENDCASE
ENDDO T
* EOF: START.PRG
```



```

*
DO WHILE .T.
*
CLEAR
* Print the menu on the screen
@ 1, 0 SAY "=====
@ 1,40 SAY "=====
@ 2, 0 SAY "||"
@ 2, 6 SAY "                P R O S P E C T   E D I T  "
@ 2,78 SAY "||"
@ 3, 0 SAY "=====
@ 3,40 SAY "=====
@ 4, 0 SAY "||"
@ 4,78 SAY "||"
@ 5, 0 SAY "||"
@ 5,78 SAY "||"
@ 6, 0 SAY "||"
@ 6,78 SAY "||"
@ 7, 0 SAY "||"
@ 7,78 SAY "||"
@ 8, 0 SAY "||"
@ 8,78 SAY "||"
@ 9, 0 SAY "||"
@ 9,78 SAY "||"
@ 10, 0 SAY "||"
@ 10,78 SAY "||"
@ 11, 0 SAY "||"
@ 11,78 SAY "||"
@ 12, 0 SAY "||"
@ 12,78 SAY "||"
@ 13, 0 SAY "||"
@ 13,78 SAY "||"
@ 14, 0 SAY "||"
@ 14,78 SAY "||"
@ 15, 0 SAY "||"
@ 15,78 SAY "||"
@ 16, 0 SAY "||"
@ 16,78 SAY "||"
@ 17, 0 SAY "=====
@ 17,40 SAY "=====
@ 5,25 SAY " 0. EXIT to MAIN MENU"
@ 7,25 SAY " 1. EDIT a Specific Record"
@ 9,25 SAY " 2. EDIT a Cadre Reps Records  "
@ 11,25 SAY " 3. EDIT entire file"
STORE "                " TO L_Name
STORE 10 TO selectnum
DO WHILE selectnum < 0 .OR. selectnum > 5
  STORE " " TO select
  @ 17,33 SAY " select : : "
  @ 17,42 GET select PICTURE "#"
  READ
  STORE VAL(select) TO selectnum
ENDDO
*

```

```

*
DO CASE
  * Exit to the main menu
  CASE selectnum= 0
    SET DELIMITER ON
    SET BELL ON
    SET TALK ON
    SET DELETED OFF
    CLEAR ALL
    RETURN
  CASE selectnum= 1
    * Edit a specific record
    STORE SPACE(20) TO L_NAME
    STORE SPACE(10) TO F_NAME
    STORE SPACE(2) TO MI
    @ 13,10 SAY "ENTER PROSPECT'S LAST NAME:           :"
    @ 13,38 GET L_Name PICTURE 'XXXXXXXXXXXXXXXXXXXXXX'
    @ 14,10 SAY "ENTER PROSPECT'S FIRST NAME :           :"
    @ 14,39 GET F_Name PICTURE 'XXXXXXXXXX'
    @ 15,10 SAY "ENTER PROSPECT'S MIDDLE INITIAL : :      "
    @ 15,43 GET MI PICTURE 'XX'
    READ
    USE PROSPECT.DBF
    LOCATE FOR L_Name = LAST_NAME .AND. F_Name =
      FIRST_NAME .AND. MI = MIDDLE_INI
    IF FOUND()
      SET FORMAT TO ROTCFORM
      EDIT
      USE
    ELSE
      CLEAR
      @ 14,20 SAY "Record Not Located in Prospect
        Database"
      WAIT SPACE(20)+"      PRESS ANY KEY TO CONTINUE.."
      USE
    ENDIF
  CASE selectnum = 2
    * Edit a Cadre Reps records
    STORE SPACE(10) TO L_NAME
    @ 13,10 SAY "ENTER CADRE REPS LAST NAME :           :"
    @ 13,38 GET L_Name PICTURE '!!!!!!!!!!!!'
    READ
    STORE LTRIM(TRIM(L_Name)) to L_Name
    USE PROSPECT.dbf
    INDEX ON CADRE_REP + LAST_NAME + FIRST_NAME TO
      CADINDEX
    SET INDEX TO CADINDEX
    SEEK L_Name
    IF FOUND()
      SET FORMAT TO ROTCFORM
      EDIT
      USE

```

*

```
ELSE
  @ 14,20 SAY " Cadre Representative Not Found in
    Database"
  WAIT SPACE(20)+ " PRESS ANY KEY TO CONTINUE..."
  SET TALK OFF
  USE
ENDIF
CASE selectnum= 3
  * Edit entire database
  USE Prospect.dbf
  INDEX on LAST_NAME + FIRST_NAME + MIDDLE_INI to
    ALPHPROS
  SET INDEX TO ALPHPROS
  SET FORMAT TO ROTCFORM
  EDIT
  USE
ENDCASE
*
ENDDO T
* EOF: EDITER.PRG
```



```

* 18 -> Index the PROSPECT database by Source and      *
* Prospect Name and call RPTFM1.PRG to print the      *
* report. Then return to the main menu (START.PRG).  *
*
* 19 -> Index the PROSPECT database by Prospect Name and *
* call RPTFM1.PRG to print a report. Then return    *
* to the main menu (START.PRG).
*
* 20 -> Call program LABELIT.PRG to print mailing labels. *
* Then return to the main menu (START.PRG).
*
* 21 -> Call program INDIV.PRG to print a report on an *
* individual prospect. Then return to the main      *
* menu (START.PRG).
*
* 22 -> Use the CONTACT database to create a report of *
* contacts with prospects grouped by Contact Rep    *
* Name and indexed by Contact Rep and Prospect     *
* Name. This report can be directed to screen or   *
* printer and deleted records may be included or   *
* eliminated from the report.
*

```

```

*****
*****
*****

```

```

*
*
* CLEAR ALL
* CLOSE ALL
*
* DO WHILE .T.
* SET TALK OFF
* SET HEADING OFF
* SET SAFETY OFF
* SET BELL OFF
* SET DELIMITER OFF
* SET STATUS OFF
*

```

```

CLEAR
@ 1, 0 SAY "===== "
@ 1,40 SAY "===== "
@ 2, 0 SAY "||"
@ 2, 6 SAY "                P R O S P E C T   R E P O R T S"
@ 2,78 SAY "||"
@ 3, 0 SAY "===== "
@ 3,40 SAY "===== "
@ 4, 0 SAY "||"
@ 4,78 SAY "||"
@ 5, 0 SAY "||"
@ 5,78 SAY "||"
@ 6, 0 SAY "||"
@ 6,78 SAY "||"
@ 7, 0 SAY "||"

```



```

@ 7,78 SAY "|||"
@ 8, 0 SAY "|||"
@ 8,78 SAY "|||"
@ 9, 0 SAY "|||"
@ 9,78 SAY "|||"
@ 10, 0 SAY "|||"
@ 10,78 SAY "|||"
@ 11, 0 SAY "|||"
@ 11,78 SAY "|||"
@ 12, 0 SAY "|||"
@ 12,78 SAY "|||"
@ 13, 0 SAY "|||"
@ 13,78 SAY "|||"
@ 14, 0 SAY "|||"
@ 14,78 SAY "|||"
@ 15, 0 SAY "|||"
@ 15,78 SAY "|||"
@ 16, 0 SAY "|||"
@ 16,78 SAY "|||"
@ 17, 0 SAY "|||"
@ 17,78 SAY "|||"
@ 18, 0 SAY "|||"
@ 18,78 SAY "|||"
@ 19, 0 SAY "|||"
@ 19,78 SAY "|||"
@ 20, 0 SAY "|||"
@ 20,78 SAY "|||"
@ 21, 0 SAY "=====|"
@ 21,40 SAY "=====|"
@ 5,20 SAY " 10. EXIT to MAIN MENU"
@ 7,20 SAY " 11. LIST of PROSPECTS by CADRE REP"
@ 8,20 SAY " 12. LIST of PROSPECTS by LAST CONTACT DATE"
@ 9,20 SAY " 13. LIST of PROSPECTS by HIGH SCHOOL GRAD
    DATE"
@ 10,20 SAY " 14. LIST of KANSAS PROSPECTS by REGION"
@ 11,20 SAY " 15. LIST of PROSPECTS by HIGH SCHOOL"
@ 12,20 SAY " 16. LIST of PROSPECTS by CITY"
@ 13,20 SAY " 17. LIST of SCHOLARSHIP RECIPIENTS by TYPE"
@ 14,20 SAY " 18. LIST of PROSPECTS by SOURCE"
@ 15,20 SAY " 19. ALPHABETICAL LIST of ALL PROSPECTS"
@ 16,20 SAY " 20. PRINT ADDRESS LABELS"
@ 17,20 SAY " 21. INDIVIDUAL REPORT"
@ 18,20 SAY " 22. LIST of CONTACTS by CONTACT REP"
STORE 25 TO selectnum
DO WHILE selectnum < 10 .OR. selectnum > 22
    STORE " " TO select
    @ 21,33 SAY " select : : "
    @ 21,42 GET select PICTURE "###"
    READ
    STORE VAL(select) TO selectnum
ENDDO
CLEAR
*
*
```

```

DO CASE
  CASE selectnum= 10
    SET DELIMITER ON
    SET BELL ON
    SET TALK ON
    SET HEADING ON
    SET SAFETY ON
    CLEAR ALL
    RETURN

  CASE selectnum= 11
    * execute the repfm1 program indexed by cadre rep
    STORE selectnum TO reptype
    USE PROSPECT.dbf
    INDEX ON CADRE_REP + LAST_NAME + FIRST_NAME +
      MIDDLE_INI TO CADINDEX
    SET INDEX TO CADINDEX
    STORE "ROTC Prospect -- CADRE Representative Report"
      to pagehdg
    DO RPTFM1.PRG
    RETURN

  CASE selectnum = 12
    * execute the conrepl program indexed by last contact
      date
    DO conrepl.prg
    RETURN

  CASE selectnum = 13
    * execute repfm1 program indexed by high school grad
      date
    USE PROSPECT.dbf
    INDEX on STR(HS_GRAD_YR,2) + LAST_NAME + FIRST_NAME +
      MIDDLE_INI to GRADINDEX
    SET INDEX to GRADINDEX
    STORE "ROTC Prospect -- High School Graduation Year
      Report" to pagehdg
    DO rptfm1.prg
    RETURN

  CASE selectnum = 14
    * execute repfm1 program indexed by region
    USE PROSPECT.dbf
    INDEX on STR(REGION,1) + LAST_NAME + FIRST_NAME +
      MIDDLE_INI to REGINDEX
    SET INDEX to REGINDEX
    STORE "ROTC Prospect -- Region Report" to pagehdg
    DO rptfm1.prg
    RETURN

  CASE selectnum = 15
    * execute repfm1 program indexed by high school
    USE PROSPECT.dbf

```

*

```

INDEX on HI_SCHOOL + LAST_NAME + FIRST_NAME +
MIDDLE_INI to HSINDEX
SET INDEX TO HSINDEX
STORE "ROTC Prospect -- High School Report" to pagehdg
DO rptfm1.prg
RETURN

CASE selectnum = 16
* execute repfm1 program indexed by city
USE PROSPECT.dbf
INDEX on CITY + LAST_NAME + FIRST_NAME + MIDDLE_INI to
CITINDEX
SET INDEX TO CITINDEX
STORE "ROTC Prospect -- City Report" to pagehdg
DO rptfm1.prg
RETURN

CASE selectnum = 17
* execute repfm1 program indexed by scholarship types
USE PROSPECT.dbf
INDEX on SCHOLAR_AP + LAST_NAME + FIRST_NAME +
MIDDLE_INI to SCLRNDX
SET INDEX to SCLRNDX
STORE "ROTC Prospect -- Scholarship Type Report" to
pagehdg
DO rptfm1.prg
RETURN

CASE selectnum = 18
* execute repfm1 program indexed by source
USE PROSPECT.dbf
INDEX on SOURCE + LAST_NAME + FIRST_NAME + MIDDLE_INI
to SRCINDEX
SET INDEX TO SRCINDEX
STORE "ROTC Prospect -- Source Report" to pagehdg
DO rptfm1.prg
RETURN

CASE selectnum = 19
USE PROSPECT.dbf
INDEX on LAST_NAME + FIRST_NAME + MIDDLE_INI to
ALPHINDX
SET INDEX TO ALPHINDX
STORE "ROTC Prospect -- Alphabetical Report" to pagehdg
DO rptfm1.prg
RETURN

CASE selectnum = 20
* execute label program
DO labelit.prg
RETURN

```

```

CASE selectnum = 21
  * execute individul view program
  DO indiv.prg
  RETURN

CASE selectnum = 22
  * contacts report
  CLEAR
  STORE " " TO select
  STORE " " TO mdel
  @ 5,0 SAY "Output to the screen or printer? [S/P] ";
  GET select PICTURE "!"
  READ

DO CASE
CASE select = "s"
  @ 10,0 SAY " Do You Want Deleted Records Included in
  Your Report? Y/N "; GET mdel PICTURE "!"
  READ
  DO CASE
    CASE mdel = "N"
      SET DELETED ON
    CASE mdel = "y"
      SET DELETED OFF
  ENDCASE
  CLEAR
  USE CONTACT.dbf
  INDEX ON C_REP+P_L_NAME+P_F_NAME+P_M_INI to CNTINDX
  REPORT FORM CONTACT.FRM
  WAIT
CASE select = "p"
  @ 10,0 SAY " Do You Want Deleted Records Included in
  Your Report? Y/N "; GET mdel PICTURE "!"
  READ
  DO CASE
    CASE mdel = "N"
      SET DELETED ON
    CASE mdel = "y"
      SET DELETED OFF
  ENDCASE
  CLEAR
  USE CONTACT.dbf
  INDEX ON C_REP+P_L_NAME+P_F_NAME+P_M_INI to CNTINDX
  REPORT FORM CONTACT.FRM TO PRINT
  SET DEVICE TO SCREEN
  ENDCASE
  SET DELETED OFF
  RETURN
*
ENDCASE
ENDDO T

```

*

Appendix U: RPTFM1.PRF Code

```

*****
*
*                               RPTFM1.PRG
*                               by
*                               Richard B. Thompson
*                               June 8, 1988
*
*****
*
*   This program generates a format used by a majority
*   of the predesigned reports. It is called by program
*   REPORTS.PRG and returns to REPORTS.PRG upon completion.
*   The report is formatted for a 130 character line
*   and is therefore directed to a printer in condensed
*   mode. This report is not designed for screen output.
*   The user is provided the option of printing or
*   omitting records marked for deletion.
*
*****
*****
*
*
CLEAR
SET MARGIN TO 10
STORE 1 TO pagenum
STORE 90 TO line
STORE (130-LEN(pagehdg))/2 TO colhdg
@ 2, 0 SAY pagehdg
@ 2,72 SAY DTOC(DATE())
@ 3, 0 SAY "=====
@ 3,40 SAY "=====
STORE "N" TO select
STORE " " TO temp
STORE "N" TO mdel
DO WHILE select <> "Y"
    @ 7,0 SAY "Turn printer on now. Is the printer on
        (Y/N)?" ; GET select PICTURE "!"
READ
ENDDO
STORE 56 TO pagelen
*
* Ask about deleted records?
*
@ 16,0 SAY " Do You Want Deleted Records Included in Your
        Report? Y/N "; GET mdel PICTURE "!"
READ

```

*

```

DO CASE
CASE mdel = "N"
    SET DELETED ON
CASE mdel = "Y"
    SET DELETED OFF
ENDCASE
SET DEVICE TO PRINT
*
DO WHILE .NOT. EOF()
    IF line > pagelen
        EJECT
        @ 0, 1 SAY " "
        @ 1, 0 SAY "Page No."
        @ 1, 9 SAY STR(pagenum,3)
        @ 1,colhdg SAY pagehdg
        @ 1,110 SAY DTOC(date())
        *
        * Generate column headings.
        *
        @ 3, 0 SAY "Name of Prospect"
        @ 3, 30 SAY "Date"
        @ 3, 41 SAY "Military"
        @ 3, 51 SAY "Source"
        @ 3, 70 SAY "Cadre Rep"
        @ 3, 84 SAY "Contact"
        @ 3, 96 SAY "Follow"
        @ 3,106 SAY "Follow Type"
        @ 4, 5 SAY "Address"
        @ 4, 30 SAY "Gold Qst"
        @ 4, 41 SAY "Scholar"
        @ 4, 51 SAY "School"
        @ 5, 5 SAY "Phone Number"
        @ 5, 30 SAY "Region"
        @ 5, 41 SAY "Grad Yr"
        @ 5, 51 SAY "Comments"
        @ 6, 0 SAY "*****"
        @ 6, 20 SAY "*****"
        @ 6, 40 SAY "*****"
        @ 6, 60 SAY "*****"
        @ 6, 80 SAY "*****"
        @ 6,100 SAY "*****"
        @ 6,120 SAY "*****"
        STORE pagenum+1 TO pagenum
        STORE 8 TO line
    ENDIF
    *
    * Print detail line.
    *
    @ line, 0 SAY TRIM(LAST_NAME)+", "+TRIM(FIRST_NAME)+"
        "+MIDDLE_INI
    @ line, 30 SAY ENTRY_DATE
    *

```

*

```

IF MIL_EXPER > "      "
  @ line,41 SAY MIL_EXPER
ELSE
  @ line,41 SAY "...."
ENDIF
*
IF SOURCE > "  "
  @ line,51 SAY SOURCE
ELSE
  @ line,51 SAY ".."
ENDIF
*
IF CADRE_REP > "          "
  @ line,70 SAY CADRE_REP
ELSE
  @ line,70 SAY "....."
ENDIF
*
IF DTOC(LST_CON_DT) > "          "
  @ line,84 SAY LST_CON_DT
ELSE
  @ line,84 SAY "....."
ENDIF
*
IF DTOC(NXT_FOL_DT) > "          "
  @ line,96 SAY NXT_FOL_DT
ELSE
  @ line,96 SAY "....."
ENDIF
*
IF NXT_FOL_TP > "          "
  @ line,106 SAY NXT_FOL_TP
ELSE
  @ line,106 SAY "....."
ENDIF
*
STORE line+1 TO line
@ line,2 SAY STREET
*
IF IN_GOLD_QS = "Y"
  @ line,30 SAY "Yes"
ELSE
  @ line,30 SAY "No"
ENDIF
*
@ line,41 SAY SCHOLAR_AP
@ line,51 SAY HI_SCHOOL
STORE line+1 TO line

```

```

IF zip > 0
  @ line,2 SAY TRIM(CITY) + ", " + STATE + " " +
    LTRIM(TRIM(STR(ZIP,10)))
ELSE
  @ line,2 SAY TRIM(CITY) + ", " + STATE
ENDIF
*
IF REGION > 0
  @ line,30 SAY REGION
ELSE
  @ line,30 SAY "... "
ENDIF
*
IF HS_GRAD_YR > 0
  @ line,41 SAY HS_GRAD_YR
ELSE
  @ line,41 SAY "... "
ENDIF
*
@ line,51 SAY SUBSTR(COMMENTS,1,75)
STORE line + 1 TO line
*
IF AREA_CODE > 0
  @ line,2 SAY "(" + STR(AREA_CODE,3) + ")"
ELSE
  @ line,2 SAY "....."
ENDIF
*
IF PHONE_NUM > 0
  @ line,8 SAY PHONE_NUM PICTURE '999-9999'
ELSE
  @ line,8 SAY "....."
ENDIF
*
STORE line+2 TO line
SKIP
ENDDO
STORE line+2 TO line
@ line+1,0 SAY " "
SET DEVICE TO SCREEN
RELEASE ALL
SET TALK ON
SET BELL ON
SET DELETED OFF
SET MARGIN TO 0
RETURN
* END OF REPORTS.PRG

```



```

READ
SET PRINT ON
STORE 52 TO pagelen
SET PRINT ON
*
DO WHILE align = "N"
? "<<<<<<----- TEST LABEL ----->>>>>>"
? "<<<<<<                                     >>>>>>"
? "<<<<<<                                     >>>>>>"
? "<<<<<<                                     >>>>>>"
? "<<<<<<----- TEST LABEL ----->>>>>>"
? " " "
SET PRINT OFF
CLEAR
@ 8, 0 SAY "Align the labels now"
@ 10, 0 SAY "Was the test label aligned? ";
GET align PICTURE "!"
READ
SET PRINT ON
ENDDO
STORE "N" TO align
DO WHILE count < (sets+1)
USE prospect.dbf
*
DO WHILE .NOT. EOF()
?
? " " " "+TRIM(TITLE)+" " "+TRIM(FIRST_NAME)+"
" "+TRIM(MIDDLE_INI)+" " "+TRIM(LAST_NAME)
? " " "+STREET
IF zip > 0
? " " " "+TRIM(CITY)+" " "+TRIM(STATE)+"
" "+LTRIM(TRIM(STR(ZIP,10)))
ELSE
? " " " "+TRIM(CITY)+" " "+TRIM(STATE)
ENDIF
?
?
SKIP
ENDDO
STORE count+1 TO count
ENDDO
*
? " " "
SET DEVICE TO SCREEN
SET PRINT OFF
SET DELETED OFF
RELEASE ALL
SET TALK ON
SET BELL ON
RETURN
* EOF: LABELIT.PRG

```

*

Appendix W: FOLLOWUP.PRG Code

```
*****
*                                     FOLLOWUP.PRG                               *
*                                     by                                           *
*                                     Richard B. Thompson                          *
*                                     June 8, 1988                                *
*****
*
*   This program is called by START.PRG when option 4
*   is selected by the user. Upon completion it returns
*   to the main menu in START.PRG.
*   The program is used to generate a menu and carry
*   out input and editing tasks as directed by the user.
*   The program accesses the FOLLOWUP database and
*   provides user with the following options:
*
*   0 -> Exit to the main menu. During the exit process,
*         this program calls a program NFOL.PRG which scans
*         the FOLLOWUP database and updates the PROSPECT
*         database with the Next Followup Date and the Next
*         Followup Type for each prospect identified in
*         both databases. A notice is printed to the
*         screen advising the user that the delay in
*         returning to the main menu is due to the update
*         process.
*
*   1 -> Input records into the FOLLOWUP database using a
*         predesigned input screen called FOLLOW1.SCR.
*
*   2 -> Edit all records in the database using the
*         predesigned input form FOLLOW1.SCR. Records are
*         displayed in record number sequence.
*
*   3 -> Edit all records in the database for a specific
*         prospect. When this option is selected, the user
*         is prompted to enter the prospects Last Name,
*         First Name and Middle Initial. Once entered, the
*         program uses the LOCATE command to locate all
*         records in the database for that specific
*         prospect. The records are displayed on the
*         screen using the FOLLOW1.SCR format.
*
*   WARNING: The FOLLOW1.SCR format provides specific
*   instructions on how to page through records and save
*   and exit the program. However, it is possible to
*   unintentionally enter blank records into the
*   database if the screen is forwarded to a blank
*   record before the data is saved. Users must hit the
*   control end key once they have entered the values in
*   the last valid record. Do not page down to the new
*   record, or you will enter a blank record into the
*   database.
*****
*
```

CLEAR ALL
 CLOSE ALL
 DO WHILE .T.
 SET TALK OFF
 SET HEADING OFF
 SET SAFETY OFF
 SET BELL OFF
 SET DELIMITER OFF
 SET STATUS OFF
 SET CONFIRM OFF

*

CLEAR

```

@ 1, 0 SAY "=====
@ 1,40 SAY "=====
@ 2, 0 SAY "|||"
@ 2, 6 SAY "          P R O S P E C T   F O L L O W - U P "
@ 2,78 SAY "|||"
@ 3, 0 SAY "=====
@ 3,40 SAY "=====
@ 4, 0 SAY "|||"
@ 4,78 SAY "|||"
@ 5, 0 SAY "|||"
@ 5,78 SAY "|||"
@ 6, 0 SAY "|||"
@ 6,78 SAY "|||"
@ 7, 0 SAY "|||"
@ 7,78 SAY "|||"
@ 8, 0 SAY "|||"
@ 8,78 SAY "|||"
@ 9, 0 SAY "|||"
@ 9,78 SAY "|||"
@ 10, 0 SAY "|||"
@ 10,78 SAY "|||"
@ 11, 0 SAY "|||"
@ 11,78 SAY "|||"
@ 12, 0 SAY "|||"
@ 12,78 SAY "|||"
@ 13, 0 SAY "|||"
@ 13,78 SAY "|||"
@ 14, 0 SAY "|||"
@ 14,78 SAY "|||"
@ 15, 0 SAY "|||"
@ 15,78 SAY "|||"
@ 16, 0 SAY "|||"
@ 16,78 SAY "|||"
@ 17, 0 SAY "=====
@ 17,40 SAY "=====
@ 7,22 SAY " 0. EXIT to MAIN MENU"
@ 9,22 SAY " 1. INPUT new FOLLOW-UP ACTION"
@ 10,22 SAY " 2. EDIT ALL FOLLOW-UP ACTIONS"
@ 11,22 SAY " 3. EDIT A SPECIFIC FOLLOW-UP ACTION"

```

*

```

STORE 10 TO selectnum
DO WHILE selectnum < 0 .OR. selectnum > 3
  STORE " " TO select
  @ 17,33 SAY " select : : "
  @ 17,42 GET select PICTURE "#"
  READ
  STORE VAL(select) TO selectnum
ENDDO
CLEAR
*
DO CASE
  CASE selectnum= 0
    * Now update the Prospect database and return to the
      main menu
    DO NFOL.PRG
    SET BELL ON
    SET TALK ON
    SET HEADING ON
    SET SAFETY ON
    SET CONFIRM OFF
    CLEAR ALL
    RETURN
  CASE selectnum= 1
    * Input new Followup action
    SET CONFIRM ON
    USE FOLLOWUP.DBF
    SET FORMAT TO FOLLOW1
    APPEND
    USE
  CASE selectnum= 2
    * Edit FOLLOWUP database
    SET CONFIRM ON
    USE FOLLOWUP.DBF
    INDEX ON P_L_NAME+P_F_NAME+P_M_INI TO FINDEX
    SET INDEX TO FINDEX
    SET FORMAT TO FOLLOW1
    EDIT
    USE
  CASE selectnum= 3
    * Edit a Specific Followup action
    SET CONFIRM ON
    USE FOLLOWUP.DBF
    STORE SPACE(20) TO L_NAME
    STORE SPACE(10) TO F_NAME
    STORE SPACE(2) TO MI
    @ 13,10 SAY "ENTER PROSPECT'S LAST NAME :           :"  

    @ 13,38 GET L_Name PICTURE 'XXXXXXXXXXXXXXXXXXXXXXX'  

    @ 14,10 SAY "ENTER PROSPECT'S FIRST NAME :           :"  

    @ 14,39 GET F_Name PICTURE 'XXXXXXXXXXXX'  

    @ 15,10 SAY "ENTER PROSPECT'S MIDDLE INITIAL : :"  

    @ 15,43 GET MI PICTURE 'XX'

```

*

```
READ
LOCATE FOR L_Name = P_L_NAME .AND. F_Name = P_F_NAME
      .AND. MI = P_M_INI
IF FOUND()
  SET FORMAT TO FOLLOW1
  EDIT
  USE
ELSE
  CLEAR
  @ 14,20 SAY " Record Not Located in Prospect
      Database"
  WAIT SPACE(20)+"      PRESS ANY KEY TO CONTINUE.."
  USE
ENDIF

ENDCASE
CLEAR
ENDDO T
* EOF: FOLLOWUP.PRG
```

Appendix X: CONTACTS.PRG Code

```
*****
*
*                               CONTACTS.PRG
*                               by
*                               Richard B. Thompson
*                               June 8, 1988
*
*****
*
*   This program is called by START.PRG when option 5 is
*   selected by the user. Upon completion it returns to
*   the main menu in START.PRG.
*   The program is used to generate a menu and carry out
*   input and editing tasks as directed by the user. The
*   program accesses the CONTACTS database and provides
*   the user with the following options:
*
*   0 -> Exit to the main menu. During the exit process,
*   this program calls a program LCON.PRG which scans
*   the CONTACTS database and updates the Last
*   Contact date field in the PROSPECT database with
*   the most recent contact date identified for the
*   prospect in the CONTACT database. A notice is
*   printed to the screen advising the user of a
*   delay in returning to the main menu due to the
*   update process.
*
*   1 -> Input a record into the CONTACTS database using a
*   predesigned input screen (CONTACT1.SCR).
*
*   2 -> Edit all records in the database for a specific
*   prospect. When this option is selected, the user
*   is prompted to enter the prospects Last Name,
*   First Name and Middle Initial. Once entered, the
*   program uses the LOCATE command to locate all
*   records in the database for that specific
*   prospect. The records are displayed on the screen
*   using the CONTACT1.SCR format. If the record is
*   not located, a message is displayed to the user
*   and the CONTACTS menu is returned.
*
*   3 -> Edit all records in the CONTACTS database for a
*   given Contact Representative. The user is
*   prompted to enter the Last Name of the contact
*   rep. Records are displayed in the CONTACT1.SCR
*   format for editing. If the record is not found,
*   the screen is cleared, a not found message is
*   displayed and the user is returned to this
*   programs menu.
*
*   4 -> Edit all records in the database using the
*   predesigned input form CONTACT1.SCR. Records are
*   displayed alphabetically by prospect name.
*
*
*                               X-1
```

```

* WARNING: The CONTACT1.SCR format provides specific *
* instructions on how to page through records and save *
* and exit the program. However, it is possible to *
* unintentionally enter blank records into the database *
* if the screen is forwarded to a blank record before *
* the data is saved. Users must hit the control end *
* key once they have entered the values in the last *
* valid record. Do not page down to the new record, or *
* you will enter a blank record into the database. *

```

```

*****
*****

```

```

SET STATUS OFF
SET HEADING OFF
SET SAFETY OFF
SET TALK OFF
SET BELL OFF
SET DELIMITER OFF
CLEAR ALL
CLOSE ALL

```

```

*
DO WHILE .T.
SET CONFIRM OFF

```

```

*
CLEAR

```

```

@ 1, 0 SAY "=====
@ 1,40 SAY "=====
@ 2, 0 SAY " | | "
@ 2, 6 SAY "          P R O S P E C T   C O N T A C T S "
@ 2,78 SAY " | | "
@ 3, 0 SAY "=====
@ 3,40 SAY "=====
@ 4, 0 SAY " | | "
@ 4,78 SAY " | | "
@ 5, 0 SAY " | | "
@ 5,78 SAY " | | "
@ 6, 0 SAY " | | "
@ 6,78 SAY " | | "
@ 7, 0 SAY " | | "
@ 7,78 SAY " | | "
@ 8, 0 SAY " | | "
@ 8,78 SAY " | | "
@ 9, 0 SAY " | | "
@ 9,78 SAY " | | "
@ 10, 0 SAY " | | "
@ 10,78 SAY " | | "
@ 11, 0 SAY " | | "
@ 11,78 SAY " | | "
@ 12, 0 SAY " | | "
@ 12,78 SAY " | | "

```

```

*

```



```

@ 13, 0 SAY "||"
@ 13,78 SAY "||"
@ 14, 0 SAY "||"
@ 14,78 SAY "||"
@ 15, 0 SAY "||"
@ 15,78 SAY "||"
@ 16, 0 SAY "||"
@ 16,78 SAY "||"
@ 17, 0 SAY "-----"
@ 17,40 SAY "-----"
@ 5,25 SAY " 0. EXIT to MAIN MENU"
@ 7,25 SAY " 1. INPUT a CONTACT"
@ 8,25 SAY " 2. EDIT CONTACTS by PROSPECT NAME"
@ 9,25 SAY " 3. EDIT CONTACTS by CONTACT REP NAME"
@ 10,25 SAY " 4. EDIT ALL CONTACTS in DATABASE"
STORE " " TO L_Name
STORE 10 TO selectnum
DO WHILE selectnum < 0 .OR. selectnum > 4
  STORE " " TO select
  @ 17,33 SAY " select : : "
  @ 17,42 GET select PICTURE "#"
  READ
  STORE VAL(select) TO selectnum
ENDDO
*
*
DO CASE
CASE selectnum= 0
  * Now update the Prospect database with contact
  changes/entries
  DO LCON.PRG
  SET DELIMITER ON
  SET BELL ON
  SET TALK ON
  SET DELETED OFF
  SET CONFIRM OFF
  CLEAR ALL
  RETURN
CASE selectnum = 1
  * enter a new contact
  SET CONFIRM ON
  USE CONTACT.dbf
  SET FORMAT TO CONTACT1
  APPEND
  USE
CASE selectnum = 2
  * Edit Contacts by Prospects name
  SET CONFIRM ON
  STORE SPACE(20) TO L_Name
  STORE SPACE(10) TO F_Name
  STORE SPACE(2) TO M_Name

```

```

@ 13,10 SAY "ENTER PROSPECT'S LAST NAME      :      :"
@ 13,38 GET L_Name PICTURE 'XXXXXXXXXXXXXXXXXXXXX'
@ 14,10 SAY "ENTER PROSPECT'S FIRST NAME     :      :"
@ 14,39 GET F_Name PICTURE 'XXXXXXXXXXXXX'
@ 15,10 SAY "ENTER PROSPECT'S MIDDLE INITIAL :      :"
@ 15,43 GET M_Name
READ
USE CONTACT.dbf
INDEX ON P_L_NAME + P_F_NAME + P_M_INI TO CONINDEX
SET INDEX TO CONINDEX
LOCATE FOR L_Name = P_L_NAME .AND. F_Name = P_F_NAME
      .AND. M_Name = P_M_INI
IF FOUND()
    SET FORMAT TO CONTACT1
    EDIT
    USE
ELSE
    CLEAR
    @ 14,20 SAY " Prospect's Name Not Found in
              Database"
    WAIT SPACE(20)+"    PRESS ANY KEY TO CONTINUE..."
    SET TALK OFF
    USE
ENDIF
CASE selectnum = 3
* Edit Contacts by Cadre Rep Name
SET CONFIRM ON
STORE SPACE(10) TO C_REP
@ 15,24 SAY "ENTER CONTACT REPS LAST NAME:      :      :"
@ 15,54 GET L_Name PICTURE '!!!!!!!!!!!!!!'
READ
STORE LTRIM(TRIM(L_Name)) to L_Name
USE CONTACT.dbf
INDEX ON C_REP + P_L_NAME + P_F_NAME + P_M_INI TO
      CADRINDX
SET INDEX TO CADRINDX
SEEK L_Name
IF FOUND()
    SET FORMAT TO CONTACT1
    EDIT
    USE
ELSE
    CLEAR
    @ 14,20 SAY " Cadre Representative Not Found in
              Database"
    WAIT SPACE(20)+"    PRESS ANY KEY TO CONTINUE..."
    SET TALK OFF
    USE
ENDIF

```

```
CASE selectnum = 4
  * Edit all contact records
  SET CONFIRM ON
  USE CONTACT.dbf
  INDEX on P_L_NAME + P_F_NAME + P_M_INI to CONALPH
  SET INDEX TO CONALPH
  SET FORMAT TO CONTACT1
  EDIT
  USE
ENDCASE
ENDDO T
* EOF: CONTACTS.PRG
```



```

READ
USE PROSPECT.DBF
LOCATE FOR L_Name = LAST_NAME .AND. F_Name =
      FIRST_NAME .AND. MI = MIDDLE_INI
IF FOUND()
  @ 15,10 SAY "TURN ON YOUR PRINTER..AND..PUT IT
      ON LINE?"
  WAIT SPACE(17)+"PRESS ANY KEY WHEN READY....."
  CLEAR
  SET DEVICE TO PRINT
  EJECT
  @ 4, 27 SAY "ARMY ROTC PROSPECT"
  @ 6, 0 SAY "ENTRY_DATE:"
  @ 6, 11 SAY DTOC(ENTRY_DATE)
  @ 8, 0 SAY "TITLE:"
  @ 8, 6 SAY TITLE
  @ 8, 12 SAY "LAST_NAME:"
  @ 8, 22 SAY LAST_NAME
  @ 8, 44 SAY "FIRST_NAME:"
  @ 8, 55 SAY FIRST_NAME
  @ 8, 67 SAY "MI:"
  @ 8, 70 SAY MIDDLE_INI
  @ 10, 0 SAY "SSN:"
  @ 10, 4 SAY STR(SSN,9)
  @ 10, 15 SAY "SEX:"
  @ 10, 19 SAY SEX
  @ 10, 22 SAY "RACE:"
  @ 10, 27 SAY RACE
  @ 10, 30 SAY "DOB:"
  @ 10, 34 SAY DTOC(DOB)
  @ 10, 44 SAY "AREA CODE:"
  @ 10, 54 SAY STR(AREA_CODE,3)
  @ 10, 59 SAY "PHONE:"
  @ 10, 65 SAY STR(PHONE_NUM,7)
  @ 12, 0 SAY "STREET:"
  @ 12, 7 SAY STREET
  @ 12, 28 SAY "CITY:"
  @ 12, 33 SAY CITY
  @ 12, 54 SAY "STATE:"
  @ 12, 60 SAY STATE
  @ 12, 64 SAY "ZIP:"
  @ 12, 68 SAY STR(ZIP,9)
  @ 14, 0 SAY "HIGH_SCHOOL:"
  @ 14, 12 SAY HI_SCHOOL
  @ 14, 48 SAY "HS_GRAD_YR:"
  @ 14, 59 SAY STR(HS_GRAD_YR,2)
  @ 16, 0 SAY "SCHOOL_DISTRICT:"
  @ 16, 16 SAY STR(SCHOOL_DIS,3)
  @ 16, 22 SAY "KSU_RECRUITING_REGION:"
  @ 16, 44 SAY STR(REGION,1)
  @ 18, 0 SAY "COLLEGE_CR:"
  @ 18, 11 SAY STR(COLLEGE_CR,3)

```

```

@ 18, 16 SAY "COLLEGE_GPA:"
@ 18, 28 SAY STR(COL_GPA,4,2)
@ 18, 36 SAY "COLLEGE_MAJOR:"
@ 18, 50 SAY COL_MAJOR
@ 20, 0 SAY "MILITARY_EXPR:"
@ 20, 14 SAY MIL_EXPER
@ 20, 22 SAY "SCHOLARSHIP_APPLICATION:"
@ 20, 46 SAY SCHOLAR_AP
@ 20, 51 SAY "ADMITTED:"
@ 20, 60 SAY ADMITTED
@ 22, 0 SAY "SOURCE:"
@ 22, 7 SAY SOURCE
@ 22, 12 SAY "REFERRED_BY:"
@ 22, 24 SAY REFER_BY
@ 22, 46 SAY "INFLUENCER:"
@ 22, 57 SAY INFLUENCER
@ 24, 0 SAY "CADRE_REP:"
@ 24, 10 SAY CADRE_REP
@ 24, 22 SAY "CADET_REP:"
@ 24, 32 SAY CADET_REP
@ 24, 52 SAY "LAST_CONTACT_DATE:"
@ 24, 70 SAY DTOC(LST_CON_DT)
@ 26, 0 SAY "NEXT_FOLLOW_UP_DATE:"
@ 26, 20 SAY DTOC(NXT_FOL_DT)
@ 26, 33 SAY "NEXT-FOLLOW-UP-TYPE:"
@ 26, 53 SAY NXT_FOL_TP
@ 28, 0 SAY "IN_GOLD_QUES:"
@ 28, 14 SAY IN_GOLD_QS
@ 28, 19 SAY "DISPOSITION:"
@ 28, 31 SAY DISPOSITN
@ 28, 37 SAY "DISPOSITION_DATE:"
@ 28, 54 SAY DTOC(DISPO_DATE)
@ 30, 0 SAY "COMMENTS:"
@ 30, 9 SAY TRIM(COMMENTS)
EJECT
SET DEVICE TO SCREEN

```

```
ELSE
```

```
  CLEAR
```

```
  @ 14,20 SAY " Record Not Located in Prospect
    Database"
```

```
  WAIT SPACE(20)+"          PRESS ANY KEY TO CONTINUE.."
  USE
```

```
ENDIF
```

```
RELEASE ALL
```

```
SET TALK ON
```

```
SET BELL ON
```

```
RETURN
```

```
ENDDO T
```

```
* EOF: INDIV.PRG
```

Appendix Z: LCON.PRG Code

```

*****
*
*                               LCON.PRG                               *
*                               by                                       *
*                               Richard B. Thompson                       *
*                               June 8, 1988                             *
*
*****
*
*   This program is called by CONTACTS.PRG and returns *
*   to the main menu (START.PRG) on completion. The *
*   program uses data in the CONTACTS database to update a *
*   field in the PROSPECT database. *
*   The program searches the CONTACTS database and *
*   identifies the latest Contact Date (C_Date) for each *
*   prospect. This date is then entered in the Last *
*   Contact Date (LST_CON_DT) field of the respective *
*   record in the PROSPECT database. *
*   A message is also printed to the user advising him *
*   of the delay in returning to the main menu due to the *
*   update. *
*
*****
*****
*
SET TALK OFF
SET BELL OFF
CLEAR ALL
CLOSE ALL
CLEAR
@10,5 SAY " PLEASE WAIT WHILE I UPDATE THE PROSPECTS
DATABASE WITH NEW CONTACTS.."
USE PROSPECT
INDEX ON LAST_NAME+FIRST_NAME+MIDDLE_INI TO PROS1
SELECT 1
USE CONTACT
SELECT 2
USE PROSPECT INDEX PROS1
DO WHILE .NOT. EOF()
    STORE LTRIM(TRIM(LAST_NAME)) TO TLNAME
    STORE LTRIM(TRIM(FIRST_NAME)) TO TFNAME
    STORE LTRIM(TRIM(MIDDLE_INI)) TO TMIDDLE
    STORE CTOD(' / / ') TO TESTDATE
    IF LST_CON_DT = TESTDATE
        STORE CTOD('01/01/60') TO TEMPDATE
    ELSE
        STORE LST_CON_DT TO TEMPDATE
    ENDIF
    ENDIF
    SELECT 1
    LOCATE FOR TLNAME=LTRIM(TRIM(P_L_NAME)) .AND.
        TFNAME=LTRIM(TRIM(P_F_NAME)) .AND.
        TMIDDLE=LTRIM(TRIM(P_M_INI))

```

*

```
DO WHILE FOUND() .AND. .NOT. EOF()
  IF FOUND()
    IF TEMPDATE < C_DATE
      STORE C_DATE TO TEMPDATE
    ENDIF
    CONTINUE
  ENDIF
ENDDO
SELECT 2
IF TEMPDATE = CTOD('01/01/60')
  REPLACE LST_CON_DT WITH CTOD('00/00/00')
ELSE
  REPLACE LST_CON_DT WITH TEMPDATE
ENDIF
SKIP
ENDDO
CLEAR
CLOSE ALL
RETURN
* EOF: LCON.PRG
```


Appendix AA: NFOL.PRG Code

```

*****
*
*
*           NFOL.PRG
*           by
*           Richard B. Thompson
*           June 8, 1988
*
*****
*
*   This program is called by FOLLOWUP.PRG and returns
*   to the main menu (START.PRG) on completion.  The
*   program uses data in the FOLLOWUP database to update
*   fields in the PROSPECT database.
*   The program searches the FOLLOWUP database and
*   identifies the next uncompleted followup date
*   (FOL_UP_DAT) for each prospect.  This date and its
*   associated followup type (FOL_UP_TYP) are then used
*   to update two fields (NXT_FOL_DT) and (NXT_FOL_TP) in
*   the in the respective prospect's record in the
*   PROSPECT database.
*   The program also prints a message to the user
*   advising him of a delay in returning to the main menu
*   due to the update.
*
*****
*
SET TALK OFF
SET BELL OFF
CLEAR ALL
CLOSE ALL
CLEAR
@10,5 SAY " PLEASE WAIT WHILE I UPDATE THE PROSPECTS
          DATABASE WITH NEW FOLLOW-UPS.."
USE PROSPECT.DBF
INDEX ON LAST_NAME+FIRST_NAME+MIDDLE_INI TO PROS1
SELECT 1
USE FOLLOWUP.DBF
SELECT 2
USE PROSPECT INDEX PROS1
DO WHILE .NOT. EOF()
  STORE LTRIM(TRIM(LAST_NAME)) TO TLNAME
  STORE LTRIM(TRIM(FIRST_NAME)) TO TFNAME
  STORE LTRIM(TRIM(MIDDLE_INI)) TO TMIDDLE
  STORE CTOD('12/31/99') TO TEMPDATE
  STORE SPACE(20) TO TEMPTYPE
  SELECT 1
  LOCATE FOR TLNAME=LTRIM(TRIM(P_L_NAME)) .AND.
           TFNAME=LTRIM(TRIM(P_F_NAME))           .AND.
           TMIDDLE=LTRIM(TRIM(P_M_INI))

```

*

```
DO WHILE FOUND() .AND. .NOT. EOF()
  IF FOUND()
    IF COMPLETED = 'N'
      IF FOL_UP_DAT < TEMPDATE
        STORE FOL_UP_DAT TO TEMPDATE
        STORE FOL_UP_TYP TO TEMPTYPE
      ENDIF
    ENDIF
    CONTINUE
  ENDIF
ENDDO
SELECT 2
IF TEMPDATE = CTOD('12/31/99')
  REPLACE NXT_FOL_DT WITH CTOD('00/00/00')
ELSE
  REPLACE NXT_FOL_DT WITH TEMPDATE
ENDIF
REPLACE NXT_FOL_TP WITH TEMPTYPE
SKIP
ENDDO
CLOSE ALL
CLEAR
RETURN
* EOF: NFOL.PRG
```



```

ENDCASE
STORE mmonth - tmonth to mmonth
  IF mmonth < 0
    STORE mmonth + 12 to mmonth
    STORE myear - 1 to myear
  ENDIF
STORE SUBSTR(STR(myear,4),3,2) to myear
STORE CTOD(STR(mmonth,2)+"/01/"+myear) TO newdate
CLEAR
STORE " " TO select
@ 5,0 SAY "Output to the screen or printer? [S/P] ";
GET select PICTURE "!";

READ
DO CASE
  CASE select = "S"
    @ 10,0 SAY " Do You Want Deleted Records Included
      in Your Report? Y/N "; GET mdel PICTURE "!"

  READ
    DO CASE
      CASE mdel = "N"
        SET DELETED ON
      CASE mdel = "Y"
        SET DELETED OFF
    ENDCASE
    CLEAR
    USE PROSPECT.dbf
    INDEX ON CADRE_REP TO CADRINDX
    SET INDEX TO CADRINDX
    REPORT FORM CONREP.FRM FOR LST_CON_DT < newdate
      HEADING "Contacts Over "+STR(selectnum,1)+ "
        month old"
    WAIT
  CASE select = "P"
    @ 10,0 SAY " Do You Want Deleted Records Included in
      Your Report? Y/N "; GET mdel PICTURE "!"

  READ
    DO CASE
      CASE mdel = "N"
        SET DELETED ON
      CASE mdel = "Y"
        SET DELETED OFF
    ENDCASE
    CLEAR
    USE PROSPECT.dbf
    INDEX ON CADRE_REP TO CADRINDX
    SET INDEX TO CADRINDX
    SET DEVICE TO PRINT
    REPORT FORM CONREP.FRM FOR LST_CON_DT < newdate HEADING
      "Contacts Over "+STR(selectnum,1)+" month old" TO
      PRINT

```

SET DEVICE TO SCREEN
ENDCASE
SET DELETED OFF
RETURN
* EOF: CONREP1.PRG

*

BB-3

Appendix CC

PROSPECT SYSTEM

User's Guide and Documentation

1. Introduction

The PROSPECT database system is designed to assist the Army ROTC department at Kansas State University with maintaining data and generating reports on prospective ROTC cadets. The system consists of three databases, eleven programs and numerous screen and report formats, all developed using dBase III Plus software.

The PROSPECT database is the main database for the system. It contains biographical data on prospective cadets. The structure of this database is listed at Appendix A.

The FOLLOWUP and CONTACT databases contain data on follow-up actions and a record of contacts with each prospective cadet. The structures for these databases are provided at Appendixes B and C.

The data dictionary (Appendix D) provides a detailed description of each field used in the PROSPECT system.

The PROSPECT system is user friendly. It is a menu driven system which, from a user perspective, closely resembles the format of the ROTC department's Cadet Management System. In the remaining sections of this guide, each menu and all possible selections are discussed in detail.

2. Using the PROSPECT System

A. Booting the System

To use the PROSPECT system, you must first load the dBase III Plus software. This is accomplished by inserting the dBase disk into the disk drive and entering the command "dbase" at the operating system prompt as shown below:

```
A>dbase
```

When you have successfully booted dBase III Plus, you will see the dot prompt "." on the screen. You can now start the PROSPECT system and bring up the MAIN MENU by inserting the PROSPECT disk into the B drive and entering the command "do start" as follows:

```
.do start
```

B. The Main Menu

The main menu for the prospect system offers seven alternatives:

```
=====
                        K S U R O T C P R O S P E C T S
=====
```

0. EXIT to dBase dot prompt
1. INPUT new Prospect into the database
2. EDIT PROSPECTS database
3. PRINT a REPORT
4. Enter/Edit a FOLLOW-UP action on a Prospect
5. Enter/Edit a CONTACT action with a Prospect
6. Leave dBase, EXIT to A>

```
===== select: : =====
```

- 0 -> This option allows you to EXIT to the dBase dot. You can reenter the PROSPECT system at any time from the dot by entering the command "do start" as shown in 2A above.
- 1 -> This option allows you to input new records into the PROSPECT database. Once selected, the MAIN MENU will disappear and the "Input Format" will appear on the screen. The Input Format is discussed in detail in section 2C.
- 2 -> This option allows you to edit/change the contents of an existing record. It can also be used to simply review the contents of a specific record or mark a record for deletion. When you select this option, the MAIN MENU disappears from the screen and the EDIT MENU will appear. This menu and its options are discussed in section 2D.
- 3 -> This option allows you to print one of twelve preformatted reports. When selected, the MAIN MENU disappears from the screen and the REPORTS MENU appears. This menu and its options are discussed in section 2E.
- 4 -> Select this option if you wish to enter or edit data concerning a follow-up action with a prospect. For example, you wish to send the prospective cadet a scholarship application in six

months. This data is entered into the FOLLOWUP database. When you select this option, the main menu disappears and the FOLLOW-UP MENU appears on the screen. This menu is discussed in section 2F.

- 5 -> Select this option if you wish to enter data concerning a contact with a prospective cadet. For example, on June 1, 1988, you telephoned the prospect and discussed scholarship options. This data is entered in the Contact database. When this option is selected, the main menu disappears and the CONTACT MENU appears on the screen. This menu is discussed in section 2G.
- 6 -> This option allows you to EXIT to the operating system output. Select this option when you have finished using the PROSPECT system. If you select this option and later wish to reenter the PROSPECT system, you must enter the commands "dbase" and "do start" as discussed in section 2A above.

C. Input Format

When option 1 is selected from the MAIN MENU, the MAIN MENU disappears from the screen and a two page input format is displayed as follows:

ARMY ROTC PROSPECT
=====

ENTRY DATE:

TITLE: LAST NAME: FIRST NAME: MI:

SSN: SEX: RACE: DOB: AREA CODE: PHONE:

STREET: CITY: STATE: ZIP:

HIGH SCHOOL: HS GRAD YEAR:

SCHOOL DISTRICT: KSU RECRUITING REGION:

COLLEGE CR: COLLEGE GPA: COLLEGE MAJOR:

PgDn Key - FOR PAGE 2 OF RECORD

MILITARY EXPR: SCHOLARSHIP APPLICATION: ADMITTED:
SOURCE: REFERRED BY: INFLUENCER:
CADRE REP: CADET REP:
IN GOLD QUEST: DISPOSITION: DISPOSITION DATE:
COMMENTS:

Esc KEY - Exit WITHOUT ADDING this record
Cntl End KEY - SAVE NEW RECORD and EXIT
PgUp KEY - Page 1 of THIS Record
PgDn KEY - SAVE AND INPUT Another Record

ENTRY DATE - This is the date the prospect is entered into the system. NOTE: THE FORMAT FOR ALL DATES IN THE PROSPECT SYSTEM IS MM/DD/YY. Example: 01/31/88.

TITLE - This field is used primarily for addressing correspondence to the prospect. It is a 4 character field. Acceptable values are: Mr, Mrs, Ms, and Miss.

LAST NAME - This is the prospective cadet's last name.
Example: Jones.

FIRST NAME - This is the prospective cadet's first name.
Example: John.

MI - This is the prospective cadet's middle initial. Note, two spaces are provided to distinguish between two cadets with the same name. Entry of one or two letters is acceptable. Example: B, or Br.

SSN - This is the prospective cadet's social security account number. This information may not always be available, and therefore, this field may be blank.
Example: 12367987.

SEX - Only two values are acceptable in this field, M for male, and F for female.

- RACE - This is a one character field. Acceptable values are:
C = White (Caucasoid)
N = Black (Negro/African)
R = Red (American Indian)
M = Yellow (Asian/Mongoloid)
X = Other
Z = Unknown
These codes are standard TRADOC race codes. This field may be left blank.
- DOB - This is the prospect's date of birth. NOTE: THE FORMAT FOR ALL DATES IN THE PROSPECT SYSTEM IS MM/DD/YY. Example: 02/23/70.
- AREA CODE - This is a three digit numeric field representing the prospect's telephone area code. Example: 913.
- PHONE - This is the prospect's seven digit telephone number. Example: 777-9856.
- STREET - This is the prospect's street address. Example: 903 Walnut Street.
- CITY - This is the city of the prospect's address. Example: Manhattan.
- STATE - This is the standard two character state code. Example: KS. Appendix E provides a list of acceptable postal service two character state codes.
- ZIP - The zip code can be entered as the standard 5 digit numeric code or the expanded nine digit code. Example: 12345, or 987654444.
- HIGH SCHOOL - Enter the name of the prospective cadet's high school in this 30 character field. Example: Manhattan.
- HS GRAD YR - Enter the two digit year that the prospective cadet graduates, or graduated from, high school. Example: 88.
- SCHOOL DISTRICT - Enter the prospective cadet's school district number, if known. This is a 3 digit numeric field. Example: 383. This field may also be left blank.

- KSU RECRUITING REGION - Enter the numeric code 0 through 9 representing the KSU recruiting region. Example: 3. This field may be blank.
- COLLEGE CREDITS - Enter the number of college credits earned by the prospect. Range of acceptable values are from blank to 999. Example: 23.
- COLLEGE GPA - Enter the grade point average attained by the prospect in college courses. Example: 2.55. This field may be blank.
- COLLEGE MAJOR - Enter the prospect's academic major. Leave blank if undecided. Example: Engineering.
- MILITARY EXPR - Enter one of the following four character military experience codes, or leave blank:
AA-C = Active Army Current
AF-C = Air Force Current
NG-C = National Guard Current
MC-C = Marine Corps Current
AR-C = Army Reserve Current
USNC = Navy Current
AA-P = Active Army Prior Service
AF-P = Air Force Prior Service
NG-P = National Guard Prior Service
MC-P = Marine Corps Prior Service
AR-P = Army Reserve Prior Service
USNP = Navy Prior Service
OTHR = Other military experience (Explain in comment field)
- SCHOLARSHIP APPLICATION - Enter a one character code as follows, or leave blank:

4 = four year
3 = three year
2 = two year
A = Active Army
N = National Guard
O = other
- ADMITTED - This is a one character code indicating the prospect's admission status. Enter one of the following one-character codes, or leave blank:
A = Admitted to college
P = Applied for admission to college
N = Not applied for admission

SOURCE - This is a two character code indicating the source of the prospect. Enter one of the following:

IN = Internal
GQ = Gold Quest
GG = Green to Gold
UR = US Army Recruiting Command
GO = Goldminer
OO = Other

REFERRED BY - This is a 20 character field which indicates the name of the person or agency who referred the prospect to the department. Example: John Smith, 101 MI Bn, etc. This field may be left blank.

INFLUENCER - This is a 20 character field which indicates the name of the person or agency who can influence the prospect's decision to enter the ROTC program. It may be left blank. Example: John Smith.

CADRE REP - Enter the last name of the Cadre Representative assigned to the prospect. This entry will automatically be transformed into upper case. Example: BROWN.

CADET REP - Enter the last name of the Cadet Representative assigned to the prospect. This entry will automatically be transformed into upper case. Example: SMITH.

IN GOLD QUEST - Enter Y if the prospect is in the National Gold Quest system or N if he is not in the system.

DISPOSITION - This is a two character field which indicates the prospects final disposition. Enter a two character code as follows:

EN = Enrolled in the ROTC program
IN = Interested in the program
NI = Not interested in the program
NQ = Not qualified for the program
OO = Other (Explain in comments field)

DISPOSITION DATE - Enter the date the final disposition was determined. NOTE: THE FORMAT FOR ALL DATES IN THE PROSPECT SYSTEM IS MM/DD/YY. Example: 02/21/88.

COMMENTS - This is a 250 character field for open comments about the prospect.

The PgDn (Page Down) Key is used to display page 2 of the record, or to save the current record and display the input format for an additional entry.

The PgUp (Page Up) Key is used to access page 1 of the current record, or access previous records entered during the input session.

The Esc (Escape) Key allows you to exit the input session without saving the current record. This will return you to the MAIN MENU.

D. Edit Menu

When you select option 2 from the MAIN MENU, the EDIT MENU appears on the screen as follows:

```
=====
                P R O S P E C T   E D I T
=====
                0.  EXIT to MAIN MENU
                1.  EDIT a Specific Record
                2.  EDIT a Cadre Repts Records
                3.  EDIT entire file

===== select:  : =====
```

0 -> Returns you to the MAIN MENU.

1 -> This option allows you to edit a specific record in the data base. The following template is displayed on the screen;

```
ENTER PROSPECT'S LAST NAME:           :
ENTER PROSPECT'S FIRST NAME:         :
ENTER PROSPECT'S MIDDLE INITIAL:     :
```

You must enter the above data carefully or a match on the requested record will not occur.

If no match is found on the specific name, the following message is displayed on the screen.

```
Record Not Located in the Prospect Database
PRESS ANY KEY TO CONTINUE..
```

When you press a key, you are returned to the EDIT MENU.

If a match is found, the record will be displayed in the EDIT FORMAT described in section 2D.1.

- 2 -> This option allows you to edit all records assigned to a specific Cadre Representative. When you select this option, the following template is displayed on the screen:

ENTER CADRE REPS LAST NAME: _____ :

Your entry is automatically transformed into upper case letters.

If the Cadre Representative is not located in the Prospect database, the following message is displayed on the screen:

Cadre Representative Not Found in Database
PRESS ANY KEY TO CONTINUE...

When you press a key, you are returned to the EDIT MENU.

If the Cadre rep is located in the database, his records are displayed alphabetically by prospect name in the EDIT FORMAT. NOTE, records belonging to other Cadre reps will follow the specified Cadre rep in the display. You will not be returned to the EDIT MENU until the Esc or Ctrl End Keys are pressed, or until the last record in the database is displayed.

- 3 -> This option allows you to edit all records in the Prospect database. The records are displayed alphabetically by prospect name in the EDIT FORMAT.

D.1. Edit Format

Records are displayed for edit in the following format:

- LAST NAME - This is the prospective cadet's last name.
Example: Jones.
- FIRST NAME - This is the prospective cadet's first name.
Example: John.
- MI - This is the prospective cadet's middle initial. Note, two spaces are provided to distinguish between two cadets with the same name. Entry of one or two letters is acceptable. Example: B, or Br.
- SSN - This is the prospective cadet's social security account number. This information may not always be available, and therefore, this field may be blank.
Example: 12367987.
- SEX - Only two values are acceptable in this field, M for male, and F for female.
- RACE - This is a one character field. Acceptable values are:
C = White (Caucasoid)
N = Black (Negro/African)
R = Red (American Indian)
M = Yellow (Asian/Mongoloid)
X = Other
Z = Unknown
These codes are standard TRADOC race codes. This field may be left blank.
- DOB - This is the prospect's date of birth. NOTE: THE FORMAT FOR ALL DATES IN THE PROSPECT SYSTEM IS MM/DD/YY. Example: 02/23/70.
- AREA CODE - This is a three digit numeric field representing the prospect's telephone area code. Example: 913.
- PHONE - This is the prospect's seven digit telephone number.
Example: 777-9856.
- STREET - This is the prospect's street address. Example: 903 Walnut Street.
- CITY - This is the city of the prospect's address. Example: Manhattan.
- STATE - This is the standard two character state code.
Example: KS. Appendix E provides a list of acceptable postal service two character state codes.

- ZIP - The zip code can be entered as the standard 5 digit numeric code or the expanded nine digit code.
Example: 12345, or 987654444.
- HIGH SCHOOL - Enter the name of the prospective cadet's high school in this 30 character field. Example: Manhattan.
- HS GRAD YR - Enter the two digit year that the prospective cadet graduates, or graduated from, high school. Example: 88.
- SCHOOL DISTRICT - Enter the prospective cadet school district number, if known. This is a 3 digit numeric field. Example: 383. This field may be left blank.
- KSU RECRUITING REGION - Enter the numeric code 0 through 9 representing the KSU recruiting region. Example: 3. This field may be blank.
- COLLEGE CREDITS - Enter the number of college credits earned by the prospect. Range of acceptable values are from blank to 999. Example: 23.
- COLLEGE GPA - Enter the grade point average attained by the prospect in college courses. Example: 2.55. This field may be blank.
- COLLEGE MAJOR - Enter the prospect's academic major. Leave blank if undecided. Example: Engineering.
- MILITARY EXPR - Enter one of the following four character military experience codes, or leave blank:
AA-C = Active Army Current
AF-C = Air Force Current
NG-C = National Guard Current
MC-C = Marine Corps Current
AR-C = Army Reserve Current
USNC = Navy Current
AA-P = Active Army Prior Service
AF-P = Air Force Prior Service
NG-P = National Guard Prior Service
MC-P = Marine Corps Prior Service
AR-P = Army Reserve Prior Service
USNP = Navy Prior Service
OTHR = Other military experience (Explain in comments field)

SCHOLARSHIP APPLICATION: Enter a one character code as follows, or leave blank:

4 = four year
3 = three year
2 = two year
A = Active Army
N = National Guard
O = other

ADMITTED - This is a one character code indicating the prospect's admission status. Enter one of the following one-character codes, or leave blank.

SOURCE - This is a two character code indicating the source of the prospect. Enter one of the following:

IN = Internal
GQ = Gold Quest
GG = Green to Gold
UR = US Army Recruiting Command
GO = Goldminer
OO = other

REFERRED BY - This is a 20 character field which indicates the name of the person or agency who referred the prospect to the department. Example: John Smith, 101 MI Bn, etc. This field may be left blank.

INFLUENCER - This is a 20 character field which indicates the name of the person or agency who can influence the prospect's decision to enter the ROTC program. It may be left blank. Example: John Smith.

CADRE REP - Enter the last name of the Cadre Representative assigned to the prospect. This entry will automatically be transformed into upper case. Example: BROWN.

CADET REP - Enter the last name of the Cadet Representative assigned to the prospect. This entry will automatically be transformed into upper case. Example: SMITH.

LAST CONTACT DATE - This is the date that the prospect was last contacted by a representative from the ROTC department. NOTE: This field is automatically updated by a software program and it will change each time a contact with the prospect is entered in the CONTACT database. The format for the field is MM/DD/YY. DO NOT EDIT THIS FIELD. To change this field, select option 5 from the MAIN MENU.

NEXT FOLLOW-UP DATE - This is a suspense date for the next contact with the prospect. NOTE: This field is automatically updated by a software program and it will change each time a new Follow-up date is entered in the FOLLOWUP database, or when a follow-up action is marked completed in the FOLLOWUP database. The format for the field is MM/DD/YY. DO NOT EDIT THIS FIELD. To change this field, select option 4 in the MAIN MENU.

NEXT FOLLOW-UP TYPE - This is the type of follow-up action required on the next follow-up date. NOTE: This field is automatically updated by a software program and it will change each time a new Follow up date is entered in the FOLLOWUP database, or when a follow-up action is marked completed in the FOLLOWUP database. This is a 20 character field. DO NOT EDIT THIS FIELD. To change this field, select option 4 in the MAIN MENU.

IN GOLD QUEST - Enter Y if the prospect is in the National Gold Quest system or N if he is not in the system.

DISPOSITION - This is a two character field which indicates the prospects final disposition. Enter a two character code as follows:

EN = Enrolled in the ROTC program
IN = Interested in the program
NI = Not interested in the program
NQ = Not qualified for the program
OO = Other (Explain in comments field)

DISPOSITION DATE - Enter the date the final disposition was determined. NOTE: THE FORMAT FOR ALL DATES IN THE PROSPECT SYSTEM IS MM/DD/YY. Example: 02/21/88.

COMMENTS - This is a 250 character field for open comments about the prospect.

The PgDn (Page Down) Key is used to display page 2 of the record, or to save the current record and display the next record to be edited.

The PgUp (Page Up) KKey is used to access page 1 of the current record, or access previously edited records in the database.

The Esc (Escape) Key allows you to edit the editing sessions without saving the current changes. You will return to the EDIT MENU.

Pressing the Ctrl (Control) and End Keys simultaneously will save your changes and return you to the EDIT MENU.

Pressing the Ctrl (Control) and U Keys simultaneously will mark the current record for deletion. If the record is already marked for deletion, pressing these keys will remove the deletion mark, or undelete/reactivate the record.

E. Reports

When you select option 3 from the main menu, a REPORTS menu is displayed on the screen as follows:

```
=====
                P R O S P E C T   R E P O R T S
=====
```

- 10. EXIT to MAIN MENU
- 11. LIST of PROSPECTS by CADRE REP
- 12. LIST of PROSPECTS by LAST CONTACT DATE
- 13. LIST of PROSPECTS by HIGH SCHOOL GRAD DATE
- 14. LIST of KANSAS PROSPECTS by REGION
- 15. LIST of PROSPECTS by HIGH SCHOOL
- 16. LIST of PROSPECTS by CITY
- 17. LIST of SCHOLARSHIP RECIPIENTS by TYPE
- 18. LIST of PROSPECTS by SOURCE
- 19. ALPHABETICAL LIST of ALL PROSPECTS
- 20. PRINT ADDRESS LABELS
- 21. INDIVIDUAL REPORT
- 22. LIST of CONTACTS by CONTACT REP

```
===== select:      : =====
```

- 10 -> Allows you to EXIT to the MAIN MENU.
- 11 -> Prints a standard report sorted by the Cadre Representative's last name. Records with no assigned Cadre Rep are printed first. See Appendix F.
- 12 -> Provides a report of prospective cadets not contacted within the past 30, 60, or 90 days. When you select this report, the REPORTS MENU disappears and a screen displaying the 30, 60, or 90 day option is displayed:

Make Your Selection

1. Last Contact greater than 30 days
2. Last Contact greater than 60 days
3. Last Contact greater than 90 days

This report also offers you the option of directing the report to the screen or printer. See Appendix G.

- 13 -> Prints a standard report sorted by high school graduation year. Records with a blank graduation year are printed first. See Appendix H.
- 14 -> Prints a standard report sorted by recruiting region. See Appendix I.
- 15 -> Prints a standard report sorted by high school name. See Appendix J.
- 16 -> Prints a standard report stored by city. See Appendix K.
- 17 -> Prints a standard report sorted by scholarship type. See Appendix L.
- 18 -> Prints a standard report sorted by source. See Appendix M.
- 19 -> Prints a standard report sorted alphabetically by prospect name. See Appendix N.
- 20 -> Prints one or more sets of address labels. See Appendix O.
- 21 -> Prints a report showing all data in the PROSPECT database on a specified individual. When you select this option, you are prompted to enter the prospect's last name, first name, and middle initial. If the prospect is found in the database, a report is printed (See Appendix P). If no match is found, an appropriate message is displayed on the screen and then you are returned to the MAIN MENU.
- 22 -> Prints a report from the CONTACT database of all contacts made by all contact representatives. You can direct this report to the screen or printer (See Appendix Q).

All of the above reports offer you the option of including or omitting deleted records in the report. The default option is "NO".

If you try to print a report without putting the printer on line, you will receive an error message as follows:

Printer not ready.

Called from - Reports.prg

Called from - Start.prg

Cancel, Ignore, or Suspend (C, I, or S)

When this occurs, you should turn the printer on, put it on line, and then enter "I". This will resume printing of the report. If you enter C or S, you will be returned to the dBase dot and you must then type "do start" to return to the MAIN MENU.

Once a report is displayed or printed, you are automatically returned to the MAIN MENU.

F. Follow-up Actions

Follow-up actions are entered/edited in the FOLLOWUP database by selecting option 4 from the MAIN MENU. When option 4 is entered, the FOLLOW-UP MENU is displayed on the screen as follows:

```
=====
      P R O S P E C T   F O L L O W - U P
=====
      0.  EXIT to MAIN MENU
      1.  INPUT new FOLLOW-UP ACTION
      2.  EDIT ALL FOLLOW-UP ACTIONS
      3.  EDIT A SPECIFIC FOLLOW-UP ACTION

===== select:  : =====
```

0 -> Allows you to return to the MAIN MENU. When you select this option, a program is run to update each record in the PROSPECT database with the earliest non completed followup date and its associated followup type. This action results in a short delay in returning to the MAIN MENU. You are informed of the delay by a message displayed briefly on the screen.

1 -> This option allows you to input a new Follow-up action. Once selected, the FOLLOWUP MENU disappears

and the Followup Screen format is displayed. This format and appropriate entries are described in section 2F.1.

- 2 -> This option allows you to edit all records in the FOLLOWUP database. The records are displayed alphabetically by prospect name in the Followup Screen format.
- 3 -> This option allows you to edit all followup actions associated with a specific prospect name. When this option is selected, you are prompted to enter the last name, first name, and middle initial of the prospect as follows:

```
ENTER PROSPECT'S LAST NAME:           :  
ENTER PROSPECT'S FIRST NAME:         :  
ENTER PROSPECT'S MIDDLE INITIAL:     :
```

The FOLLOWUP database is then searched for all records associated with the specified name. The found records are displayed in the Followup Screen format. If no records are found, a message is printed to the screen, and you are returned to the FOLLOWUP MENU.

F.1. Follow-up Screen Format

Followup actions are entered and edited in the FOLLOWUP database using the following screen format:

FOLLOW-UP ENTRY =====

PROSPECTS LAST NAME XXXXXXXXXXXXXXXXXXXXX	PROSPECTS FIRST NAME XXXXXXXXXX	PROSPECTS MIDDLE INITIAL XX
DATE OF FOLLOW-UP MM/DD/YY	TYPE OF FOLLOW-UP ACTION #####	COMPLETED Y/N N
Esc Key	- Exit WITHOUT SAVING	
Ctrl U Key	- DELETE THIS ENTRY	
Ctrl End Key	- SAVE and EXIT	
PgDn Key	- NEXT ENTRY	
PgUp Key	- PREVIOUS ENTRY	

PROSPECTS LAST NAME - This is the prospective cadet's last name. Example: Jones.

PROSPECTS FIRST NAME - This is the prospective cadet's first name. Example: John.

PROSPECTS MIDDLE INITIAL - This is the prospective cadet's middle initial. Note, two spaces are provided to distinguish between two cadets with the same name. Entry of one or two letters is acceptable. Example: B, or Br.

DATE OF FOLLOW-UP - Enter the suspense date for the follow-up action. NOTE: THE FORMAT FOR ALL DATES IN THE PROSPECT SYSTEM IS MM/DD/YY. Example: 11/20/89.

TYPE OF FOLLOW-UP ACTION - Enter the type of action required on the associated follow-up date. Example: Scholarship letter.

COMPLETED Y/N - Enter Y if the follow-up action has been completed, and N if it is not completed.

Use the Esc Key to return to the FOLLOWUP MENU without saving changes to the current record.

The Ctrl (Control) and U Keys (depressed simultaneously) marks a record for deletion. If the record is already marked for deletion, depressing these keys will reactivate a deleted record (remove the deleted mark).

The Ctrl (Control) and End Keys (depressed simultaneously) save the current record and return you to the FOLLOWUP MENU.

The PgUp (Page Up) key displays the previous entry.

The PgDn (Page Down) key displays the next entry. WARNING: When entering data into the FOLLOWUP database, do not page down to a blank record after your last input entry. Such action will result in inserting a blank record into the file. The correct procedure is to hit the Ctrl End keys when the last valid entry is still on the screen. This will save the current input and return you to the FOLLOWUP MENU without inserting a blank record.

G. Contact Actions

Contact actions are entered/edited in the CONTACT database by selecting option 5 from the MAIN MENU. When option 5 is selected, the CONTACT MENU is displayed on the screen as follows:

Your response is automatically transformed into upper case letters. The CONTACT database is then searched for all records associated with the contact representative, and the records are displayed on the screen in the Contact Screen format. If no records are found, an appropriate message is displayed on the screen, and you are returned to the CONTACT MENU.

- 4 -> This option allows you to edit all records in the CONTACT database. When this option is selected, the records are displayed alphabetically by prospect name in the Contact Screen format.

G.1. Contact Screen Format

Contact actions are entered/edited in the CONTACT database using the following Contact Screen format:

CONTACT ENTRY
=====

PROSPECTS LAST NAME XXXXXXXXXXXXXXXXXXXX	PROSPECTS FIRST NAME XXXXXXXXXXXX	PROSPECTS MIDDLE INITIAL XX	CONTACT REMARKS
DATE OF CONTACT MM/DD/YY	CONTACT REPRESENTATIVE XXXXXXXXXX	CONTACT METHOD x	CONTACT REMARKS #####
	Esc Key	-	Exit WITHOUT SAVING
	Ctrl U Key	-	DELETE THIS ENTRY
	Ctrl End Key	-	SAVE and EXIT
	PgDn Key	-	NEXT ENTRY
	PgUp Key	-	PREVIOUS ENTRY

PROSPECTS LAST NAME - This is the prospective cadet's last name. Example: Jones.

PROSPECTS FIRST NAME - This is the prospective cadet's first name. Example: John.

PROSPECTS MIDDLE INITIAL - This is the prospective cadet's middle initial. Note, two spaces are provided to distinguish between two cadets with the same name. Entry of one or two letters is acceptable. Example: B, or Br.

DATE OF CONTACT - Enter the date of the contact with the prospect. NOTE: THE FORMAT FOR ALL DATES IN THE PROSPECT SYSTEM IS MM/DD/YY. Example: 06/20/88.

CONTACT REPRESENTATIVE - Enter the name of the person who contacted the prospective cadet. This entry is automatically transformed into upper case letters. Example: BROWN.

CONTACT METHOD - Enter a one-character code as follows, to indicate the method of contact:

P = in Person

T = by Telephone

M = by Mail

N = not Contacted

a blank is not allowed in this field.

CONTACT REMARKS - You may enter up to 30 characters of open remarks in this field. This field may also be left blank.

Use the Esc Key to return to the CONTACT MENU without saving changes to the current record.

The Ctrl (Control) and U Keys (depressed simultaneously) mark a record for deletion. If the record is already marked for deletion, depressing these keys will reactivate a deleted record (remove the deleted mark).

The Ctrl (Control) and End Keys (depressed simultaneously) save the current record and return you to the CONTACT MENU.

The PgUp (Page Up) key displays the previous entry.

The PgDn (Page Down) key displays the next entry.
WARNING: When entering data into the CONTACT database, do not page down to a blank record after your last input entry. Such action will result in inserting a blank record into the file. The correct procedure is to hit the Ctrl End keys when the last valid entry is still on the screen. This will save the current record and return you to the CONTACT MENU without inserting a blank record.

3. Conclusion

The PROSPECT system was designed specifically for personnel with minimal computer experience. As you become familiar with the system, you will undoubtedly search for short cuts to avoid wading through the PROSPECT menus.

One possible short cut is the use of dBase ASSIST or EDIT commands to enter or change data in the databases. This method is perfectly acceptable for the PROSPECT database. However, dBase commands MUST NOT be used to edit the CONTACT or FOLLOWUP databases. The programs in the

PROSPECT system which edit these databases call subroutines which automatically update fields in the PROSPECT database with data from the CONTACT and FOLLOWUP databases. If the PROSPECT system is not used to edit the CONTACT and FOLLOWUP databases, the data in the three databases will be inconsistent.

Appendix DD

Definitions

Anomaly: Situations which result in the loss of information or inconsistent information in a database. There are 13 anomalies defined formally by the standards organization. Three are of great interest: insert, delete, and maintenance.

Prime Attributes: An attribute of a relation that participates in (or as part of) any key.

Nonprime Attribute: An attribute that participates in no key.

Functional Dependency: A relationship between attributes X and Y where X functionally determines Y if the value of X determines the value of Y.

Multivalued Dependency: Refers to a 1:N relationship among attributes where the N portion of the functional mapping is composed of orthogonal disjoint attributes.

Partial Dependency: A partial dependency exists when a nonprime attribute can be functionally determined by a subset of one or more of the keys of the relation.

Transitive Dependency: A transitive dependency exists in a relation when a nonprime attribute, A, participates in the following relationship: Key \rightarrow Nonkey \rightarrow A.

Key: An attribute or set of attributes which uniquely identifies every tuple in the relation.

Foreign Key: A set of attributes in one relation say R1, which is a key in another relation say R2.

Primary Key: A key which is used as the primary storage mechanism in the physical database.

Super Key: A set of attributes which contains a key and additional attributes.

Relation: A two-dimensional table with the following properties: 1) the table entries are atomic, 2) the column entries are all of the same kind, 3) each column has a unique name, 4) the order of the column entries is immaterial, 5) no two rows in the table are identical, and 5) the order of the rows in the table is immaterial.

Schema: The logical database design which specifies the logical format of the database. The specification includes the records to be maintained, their contents, and relationships among the records.

A STUDY OF THE USE OF DOMAIN KEY
NORMAL FORM CRITERIA IN DATABASE DESIGN

by

RICHARD BRYAN THOMPSON

B.S., University of North Dakota, 1970

AN ABSTRACT OF A REPORT

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

COMPUTER SCIENCE

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1988

Abstract

Modern database theory suggests that the quality of a database design is directly related to the level of the schema's normal form. There are at least five accepted levels of normal form. Lower level normal forms possess more anomalies than the higher levels; therefore, the quality of the database generally improves as the normal form level increases.

Domain Key Normal Form is virtually free of anomalies. It is considered the highest level of normal form and is therefore the goal of many database designers. Achieving a Domain Key schema is not easy however, and there is no known mechanical method to transform a lower normal form schema into a Domain Key schema.

This study suggests a design method for achieving a Domain Key Normal Form schema. It introduces the "KSU Adhoc Method" of design and demonstrates how Domain Key Normal Form can be achieved through the application of a four step design process.

The study also presents a design and implementation of an automated database. The database was developed to support the recruiting activities of the KSU Army ROTC Department.