The Development of a Database Application and its Interface for the Computer Science Research Consortium Information System

by

John Joseph Imholz

Project submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of MASTERS OF INFORMATION SYSTEMS in
Computer Science

APPROVED:

H. Rex Harrison, Chairman

Richard E. Nance                     James O. Hicks

November, 1984
Blacksburg, Virginia
TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>II. REQUIREMENTS</td>
<td>3</td>
</tr>
<tr>
<td>General</td>
<td>3</td>
</tr>
<tr>
<td>Initial</td>
<td>4</td>
</tr>
<tr>
<td>Additional</td>
<td>4</td>
</tr>
<tr>
<td>III. DESIGN</td>
<td>7</td>
</tr>
<tr>
<td>Relations</td>
<td>7</td>
</tr>
<tr>
<td>System Interface</td>
<td>11</td>
</tr>
<tr>
<td>Main</td>
<td>12</td>
</tr>
<tr>
<td>People</td>
<td>13</td>
</tr>
<tr>
<td>Mail</td>
<td>18</td>
</tr>
<tr>
<td>List</td>
<td>19</td>
</tr>
<tr>
<td>Modify</td>
<td>19</td>
</tr>
<tr>
<td>Print and Type</td>
<td>22</td>
</tr>
<tr>
<td>IV. IMPLEMENTATION AND DESIGN CYCLE</td>
<td>25</td>
</tr>
<tr>
<td>V. EVALUATION</td>
<td>27</td>
</tr>
<tr>
<td>Designers Work</td>
<td>27</td>
</tr>
<tr>
<td>dBase II</td>
<td>28</td>
</tr>
<tr>
<td>Superman</td>
<td>30</td>
</tr>
<tr>
<td>VI. CONCLUSIONS</td>
<td>31</td>
</tr>
<tr>
<td>Future Work</td>
<td>31</td>
</tr>
<tr>
<td>dBase III</td>
<td>32</td>
</tr>
<tr>
<td>VII. REFERENCES</td>
<td>33</td>
</tr>
<tr>
<td>Appendix</td>
<td>34</td>
</tr>
<tr>
<td>A. BACKUP</td>
<td>34</td>
</tr>
<tr>
<td>B. SYSTEM CODE</td>
<td>37</td>
</tr>
<tr>
<td>Brocomm</td>
<td>38</td>
</tr>
<tr>
<td>Disperso</td>
<td>40</td>
</tr>
<tr>
<td>Command</td>
<td>Page</td>
</tr>
<tr>
<td>--------------</td>
<td>------</td>
</tr>
<tr>
<td>Getlist</td>
<td>42</td>
</tr>
<tr>
<td>Getmname</td>
<td>47</td>
</tr>
<tr>
<td>Getname</td>
<td>48</td>
</tr>
<tr>
<td>Labels</td>
<td>49</td>
</tr>
<tr>
<td>Mail</td>
<td>51</td>
</tr>
<tr>
<td>Main</td>
<td>54</td>
</tr>
<tr>
<td>MakeList</td>
<td>56</td>
</tr>
<tr>
<td>Nothing</td>
<td>58</td>
</tr>
<tr>
<td>Pack</td>
<td>59</td>
</tr>
<tr>
<td>People</td>
<td>61</td>
</tr>
<tr>
<td>Redup</td>
<td>64</td>
</tr>
<tr>
<td>Reindex</td>
<td>65</td>
</tr>
<tr>
<td>SeeCom</td>
<td>66</td>
</tr>
<tr>
<td>SeeMail</td>
<td>69</td>
</tr>
<tr>
<td>SeePerso</td>
<td>73</td>
</tr>
<tr>
<td>Unpack</td>
<td>75</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>1.</td>
<td>PEOPLE Relation Schema</td>
</tr>
<tr>
<td>2.</td>
<td>REMARKS Relation Schema</td>
</tr>
<tr>
<td>3.</td>
<td>Main Menu</td>
</tr>
<tr>
<td>4.</td>
<td>Person Screen</td>
</tr>
<tr>
<td>5.</td>
<td>Person Command Menu</td>
</tr>
<tr>
<td>6.</td>
<td>Mail Menu</td>
</tr>
<tr>
<td>7.</td>
<td>Mail Definition Menu</td>
</tr>
</tbody>
</table>
Chapter I

INTRODUCTION

During the Fall of 1983 the Virginia Tech Computer Science Research Consortium (CSRC) sent out over 900 letters and forms to alumni, business associates, and friends. This was the first step to recruit members for the Consortium. The mailing list used was stored in a SCRIPT file on a mainframe computer. After a month, the replies were received at the rate of 15 per week, for several weeks. The returning forms were filed into a 3 ring binder.

The CSRC had used a list of names and addresses on the mainframe. When the forms were returned, some indicated a change of address, or a misspelling of a name, while others indicated that another person in their organization should also be contacted. With these replies, the CSRC now had an outdated list of names and no easy way to keep it updated. It was also a tedious operation to determine which persons replied to the initial letter.

The CSRC needed an information system which could process all information for each person, and which could retrieve each person's information very quickly. For example, if a person calls the CSRC on the phone, the receiver of the call needs the information about that person immediately. The
system must manage not just the name, company, and address of a person, but also other information regarding membership type and contact type. There must also be provisions for general comments, alumni information, and donations.

This report describes the requirements, design, and implementation of the CSRC database system. The system was implemented on an IBM-PC using dBase II. The SUPERMAN methodolog was used to create a system interface in which a system user's guide would not be needed.
2.1 GENERAL

The first requirement is hardware for the system. The CSRC had at its disposal a Zenith Z-100, a microcomputer which is highly compatible with the IBM-PC. The Z-100 requirement made the system quite portable because many microcomputers are compatible with the IBM-PC.

The database system which would be used was dBase II, a relational database system produced by Ashton-Tate Inc. Again, this requirement did not seem to limit the system in any way but it did improve the transportability, since dBase II is available for many microcomputers and many operating systems.

The design methodology to be used was the Supervisory Methodology and Notation (SUPERMAN) [YUNTT84]. This methodology was chosen over the Yourdon approach of structured design [YOURE79] because it develops the system with the user interface as a primary concern. The methodology lets the user, as well as the designer, communicate using the Graphical Programming Language [YUNTT84].
2.2 **INITIAL**

The first formal list of design requirements, given in October of 1983, was as follows:

1. Logical design (data definition) of necessary relations.

2. Procedures for each separate "Case" to be handled (approximately 20 cases). Include instructions for non-database part of the procedures.

3. Ability to modify existing relations.

4. Bulk input procedure.

5. Information retrieval for use in phone calls.

6. Production of mailing labels for selected subsets of the database. Sortable by zip code, alphabetized last names, etc.

7. Ability to produce printed reports (e.g., lists of member companies or entire database) sorted on various fields.

8. Documentation of all procedures (user/operations manual and technical manual or changing data definitions and procedures).

9. Ability to produce hard copy listings of mailing lists and detailed member information.

2.3 **ADDITIONAL**

The previous list of requirements constituted the users' "wish" list. These were what the users initially thought they wanted. As the system was developed, there were additional requirements for the system. These requirements
were made over the course of the development and implementation cycle.

A major requirement for the system was that a user's guide not be necessary for daily operations. (It is necessary, however, to provide written instructions for extraordinary functions such as backing up the system and modifying the relations. These functions are explained in the appendices of this report.) The absence of a user's guide required that each function of the system be fully self-explanatory. A separate "help" screen must be provided each time the screen size inhibits the ability to describe a function on the same screen as the function itself.

For a system without a user's guide to be successful, it must meet some human-oriented requirements. For example, basic functions must be available at all times and the command for these functions should be consistent. The system must also flow in a logical manner. That is, the system should display the screens that the users expect to see. The users may still be apprehensive about the system until they find that destructive commands such as DELETE are confirmed before any action takes place.

To make the system easy to use, the commands should be as short as possible to minimize the number of keystrokes. The commands should also be cognitively direct choices from the
menu instead of having numbers to represent the commands. This requirement produces problems in choosing the command symbol for each function and keeping the commands consistent throughout the system.

The smallest number of keystrokes required to input a command is one. This means that the system immediately accepts the first key entered without even a "carriage return". The process of assigning unique single character commands to all of the functions is trivial. The problem occurs when you try to make the commands cognitively direct. For example, it was not possible to use the characters "N"ext and "P"revious for the appropriate functions, because the "P" command was already assigned to the "P"erson information function. The commands for "N"ext and "P"revious were changed to "F"orward and "B"ackward.

The users wanted two commands which could be executed from anywhere in the system. They were QUIT and EXIT. The QUIT command brings the system to the CSRC main menu. The EXIT command removes the user from the CSRC application system and dBase II and places the user back in the micro's operating system.

The final requirement was to design the system such that any future functions and information needs could be easily implemented.
Chapter III
DESIGN

The design decisions can be grouped into two categories: (1) the logical design of the database and (2) the design of the interface.

3.1 RELATIONS

To maintain data independence and integrity, the goal in designing the relations was to achieve at least third normal form [DATEC82]. The basic information needs of the CSRC formed two relations. All of the information in the system is related to the people who are CSRC contacts, motivating the PEOPLE relation. The other relation was the general REMARKS relation, which contained various annotations not fitting directly into the PEOPLE relation.

The PEOPLE relation's primary key is the person's full name. The information requirements for a person produced the schema shown in figure 1. This relation is not third normal form. There are dependencies in the person's address, and also contact name and phone. However, these are not serious variations from the basic motivation to strive for a higher normal form. The "one fact in one place" [DATEC82] design principle is followed.
<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Attribute type</th>
<th>Attribute width</th>
</tr>
</thead>
<tbody>
<tr>
<td>PERSON: NBR</td>
<td>INTEGER</td>
<td>4</td>
</tr>
<tr>
<td>TITLE</td>
<td>CHARACTER</td>
<td>5</td>
</tr>
<tr>
<td>FIRST: NAME</td>
<td>CHARACTER</td>
<td>15</td>
</tr>
<tr>
<td>LAST: NAME</td>
<td>CHARACTER</td>
<td>15</td>
</tr>
<tr>
<td>PRONOUNC3E</td>
<td>CHARACTER</td>
<td>15</td>
</tr>
<tr>
<td>NICK: NAME</td>
<td>CHARACTER</td>
<td>10</td>
</tr>
<tr>
<td>MIDDLE: NAME</td>
<td>CHARACTER</td>
<td>10</td>
</tr>
<tr>
<td>SUFFIX</td>
<td>CHARACTER</td>
<td>5</td>
</tr>
<tr>
<td>CATEGORY</td>
<td>CHARACTER</td>
<td>33</td>
</tr>
<tr>
<td>JOB: TITLE</td>
<td>CHARACTER</td>
<td>40</td>
</tr>
<tr>
<td>ORG: NAME</td>
<td>CHARACTER</td>
<td>40</td>
</tr>
<tr>
<td>MAIL3</td>
<td>CHARACTER</td>
<td>33</td>
</tr>
<tr>
<td>CITY</td>
<td>CHARACTER</td>
<td>20</td>
</tr>
<tr>
<td>STATE</td>
<td>CHARACTER</td>
<td>2</td>
</tr>
<tr>
<td>ZIP</td>
<td>CHARACTER</td>
<td>5</td>
</tr>
<tr>
<td>ZIP2</td>
<td>CHARACTER</td>
<td>4</td>
</tr>
<tr>
<td>WORK: PHONE</td>
<td>CHARACTER</td>
<td>20</td>
</tr>
<tr>
<td>MEMB: DATE</td>
<td>CHARACTER</td>
<td>8</td>
</tr>
<tr>
<td>SOUR: NAME</td>
<td>CHARACTER</td>
<td>20</td>
</tr>
<tr>
<td>GEOG: AREA</td>
<td>CHARACTER</td>
<td>1</td>
</tr>
<tr>
<td>CONT: TYPE</td>
<td>CHARACTER</td>
<td>1</td>
</tr>
<tr>
<td>MEMB: TYPE</td>
<td>CHARACTER</td>
<td>1</td>
</tr>
<tr>
<td>COMMENTS</td>
<td>CHARACTER</td>
<td>1</td>
</tr>
<tr>
<td>CHECK: ORG</td>
<td>CHARACTER</td>
<td>1</td>
</tr>
</tbody>
</table>

**Figure 1:** PEOPLE Relation Schema
Some of the information in the relations is encoded. For example, CONT_TYPE is a single character defining the contact type of a person. This was originally done to lower the number of attributes in the relation (dBase II allows a maximum of thirty two). The attribute stayed encoded because the user never has to look at the data or retrieve the data using the encoded attributes, and thus, this allowed more space to define future attributes in the relation. In the initial design, the number of attributes exceeded forty. After many iterations of implementation and design, the number of attributes dropped to twenty-four.

Another needed relation was a relation for comments, remarks, and notes about a person. This information could not be included in the PERSON relation because one person could have more than one set of comments. This also saved secondary storage, which was at a premium when the system was on the Zenith. The person's full name could have been used as the foreign key for the comments relation, but this would lead to problems in updating the name fields in the person relation. So a system-defined key PERSON:NBR was included in the REMARKS relation and the PERSON relation. This key is also used in any other added relations such as CONTRIBUTIONS and ALUMNI. The REMARKS relation schema is shown in figure 2.
<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Attribute type</th>
<th>Attribute width</th>
</tr>
</thead>
<tbody>
<tr>
<td>PERSON:NBR</td>
<td>INTEGER</td>
<td>4</td>
</tr>
<tr>
<td>DATE</td>
<td>CHARACTER</td>
<td>8</td>
</tr>
<tr>
<td>PHONE</td>
<td>CHARACTER</td>
<td>1</td>
</tr>
<tr>
<td>LETTER</td>
<td>CHARACTER</td>
<td>1</td>
</tr>
<tr>
<td>OTHER</td>
<td>CHARACTER</td>
<td>1</td>
</tr>
<tr>
<td>TEXT1</td>
<td>CHARACTER</td>
<td>75</td>
</tr>
<tr>
<td>TEXT2</td>
<td>CHARACTER</td>
<td>75</td>
</tr>
<tr>
<td>TEXT3</td>
<td>CHARACTER</td>
<td>75</td>
</tr>
<tr>
<td>TEXT4</td>
<td>CHARACTER</td>
<td>75</td>
</tr>
</tbody>
</table>

**Figure 2**: REMARKS Relation Schema
The attributes TEXT1 through TEXT4 are for the notes or comments for the person. The three one-character attributes were to indicate whether the comment was a phone message, a response from a letter, or another type of message. The date is initialized when the comment is created, and can be modified. The only link to the person who owns this comment is the PERSON:NBR attribute. This makes the individual remarks independent from any information in the PEOPLE relation. If other information about people in the system is needed, then the PERSON:NBR should be used as a foreign key.

3.2 SYSTEM-INTERFACE-

Because a system user's guide was not developed, instructions on each screen were carefully worded as to what inputs can be made and the system responds to each valid command. There are also several "Information" screens which explain the current process being performed. An important feature on every screen is the option to QUIT the CSRC system or EXIT the dBase II program. The system also accepts QUIT any time as a user response.

Before the explanation of each system screen and its functions, there should be a brief general overview of the system control structure. The main system menu is displayed
when the system is first invoked. From this screen the users can go into any one of three subsystems: the PEOPLE, MAIL, and REMARKS systems. The PEOPLE and the MAIL systems are composed of a network of screens which either display information retrieved from the database or the current list of commands.

Although the system is designed hierarchically, in the MAIL and PEOPLE subsystems the user has the ability to move up and down more than one level with just one command. This allows the user to execute many functions of the system from any command menu. Otherwise the user would have to back out many menus to quit or to proceed down another set of menus to get to the desired function.

The next few sections discuss the displays of the CSRC system and what the system is doing and can do at each screen.

3.3 MAIN

The main menu (figure 3) is immediately displayed when the user enters the CSRC system (in dBase II type "DO MAIN"; in DOS type "CSRC"). While this menu is displayed, the CSRC system is waiting for the user to decide on either the PEOPLE, MAIL, or REMARKS subsystem. Each is entered by typing the corresponding menu choice character. When the
user returns from any of the three subsystems, the MAIN screen is displayed again. As with every screen in the system, the user can type "Q" to quit or "X" to exit.

3.3.1 People

When a user enters the PEOPLE subsystem, the system first displays a criterion selection screen, which asks for search criteria. The user can enter the person's last name and first name or any prefix of the concatenation of the two. The user can also specify the company name. These two items, name and company name, can be entered in any combination.

To input a new person into the database, the user just types the member's name into the criterion selection screen. The system requests confirmation before entering the new member. If any part of the company name is given and the search criteria are not matched, the user is informed and the criteria screen is displayed again with the same inputs which were given by the user. The user can modify the original inputs or enter "Q" or "X".

Once a person is found with the criteria or the new entry is confirmed, the person screen is displayed (figure 4). This screen shows all the attributes of a person and the values of the current tuple (person record). In the case of
CSRC DATABASE MAIN MENU

TYPE THE LETTER OF YOUR CHOICE:

P - ENTER, MODIFY, OR SEE INFORMATION FOR A PERSON
M - PRODUCE MAILING LABELS
R - BROWSE REMARKS, PHONE MESSAGES, ETC.
Q - EXIT FROM CSRC SYSTEM (TYPE "DO MAIN" TO RETURN)
X - EXIT FROM DBASE SYSTEM (TYPE "DBASE MAIN" TO RETURN)

Figure 3: Main Menu
an input the values are all blanks. The screen is a full screen form-filling process. All of the data can be modified or given as input except for the system generated PERSON: NBR and the cursor can be moved to any attribute value. The person command menu screen (figure 5) is displayed by entering the PqDn key (or CTRL-C if not on IBM-PC). This screen displays the valid commands for this person.

Three commands are available for moving around in the PERSON relation. They are "B"ackward, "F"orward, and "C"urrent. These commands move one tuple forward ("F"), move one tuple backward ("B") and stays at the current tuple ("C"). The PEOPLE relation is ordered alphabetically by either the peoples' last names or by the company names. If only the name of a person is given then the PEOPLE relation is ordered by last name. The PEOPLE relation is ordered by company name otherwise.

The "P" command takes the user to the criteria screen again, with all the current inputs displayed. The "R"emarks command allows the user to look at the remarks command if the person has any remarks. If the remark field is not marked on the person screen, then that person does not have any remarks and the "R" command is neither displayed nor valid. Again the user can input "Q" or "X" to quit or exit.
PERSON SCREEN: ENTER OR MODIFY DATA. Press to NEXT MENU
PERSON NUMBER = 1001

<table>
<thead>
<tr>
<th>TITLE</th>
<th>Mr.</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIRST NAME</td>
<td>John</td>
</tr>
<tr>
<td>LAST NAME</td>
<td>Imholz</td>
</tr>
<tr>
<td>PRONOUNCE</td>
<td></td>
</tr>
<tr>
<td>NICKNAME</td>
<td></td>
</tr>
<tr>
<td>MID NAME</td>
<td>Joseph</td>
</tr>
<tr>
<td>SUFFIX</td>
<td></td>
</tr>
<tr>
<td>JOB TITLE</td>
<td>Programmer</td>
</tr>
<tr>
<td>COMPANY NAME</td>
<td>CSRC</td>
</tr>
<tr>
<td>STREET</td>
<td>6200 B Terrace View</td>
</tr>
<tr>
<td>CITY,STATE,ZIP</td>
<td>Blacksburg</td>
</tr>
<tr>
<td>CATEGORY</td>
<td></td>
</tr>
<tr>
<td>WORK PHONE</td>
<td>(703) 961-1234-(0000)</td>
</tr>
<tr>
<td>MEMBERSHIP DATE</td>
<td>01/02/84</td>
</tr>
<tr>
<td>ORG. MEMB CLASS</td>
<td>INDIVIDUAL</td>
</tr>
<tr>
<td>CONTACT</td>
<td>PRIMARY</td>
</tr>
</tbody>
</table>
| GEOG. AREA    | TECH:
|              | OTHER VA: |
|              | SECOND |
|              | LOCAL: |
|              | OUT OF VA: |
|              | CONTRIBUTOR X: |
|              | STEERING |
|              | NO VA: |
|              | NON-MEMBER |

Figure 4: Person Screen
PERSON COMMAND MENU

TYPE THE LETTER OF YOUR CHOICE: :

C - <DEFAULT> CURRENT PERSON INFORMATION SCREEN
P - ENTER, MODIFY, OR SEE INFORMATION FOR ANOTHER PERSON
B - MOVE BACKWARD ONE PERSON ***** RELATION IS ORDERED
F - MOVE FORWARD ONE PERSON ***** BY LAST NAME
L - LIST OTHER EMPLOYEES OF SAME COMPANY
R - ENTER REMARKS, PHONE CALLS, LETTERS, ETC..

Q - QUIT TO MAIN MENU
X - EXIT FROM SYSTEM (TYPE "DBASE MAIN" TO RETURN)

Figure 5: Person Command Menu
3.3.1.1 Remarks

The remarks command displays the first comment for the current person. If the person does not have any comments a blank form is displayed. The date field for the remark defaults to the current date (if new) but this date can be modified. Four lines of unformatted text are available for the user's input. After input or modification, the user enters PgDn to display the command menu.

The menu contains the usual movement commands "B"ackwards, "F"orwards, and "C"urrent. There is also the ability to "A"dd and "D"elete remarks. The network view of the system is demonstrated by the two commands "P"eople and "R"eturn. The "P"eople command takes the user back to the criteria screen with the last set of given inputs displayed. The "R"eturn command takes the user back to the current person command menu.

3.4 MAIL

The MAIL subsystem is used to create or print a mailing list. This subsystem can also be used for the general retrieval of persons using criteria other than the name and company of a person. The basic scenario for the subsystem is to define the search criteria and then have the results printed or typed to the screen. The MAIL menu (figure 6)
has the following valid commands: "L"ist, "M"odify, "P"rint, "T"ype, "Q"uit, and e"X"it. The "M"odify and output commands are discussed in the following sections.

3.4.1 List-

The "L"ist command lists the current names of mailing lists having existing definitions and shows the number of people in the corresponding lists. Each mailing list is a relation with attributes for the names and addresses of retrieved tuples from the PEOPLE relation. These relations are created using a corresponding mail list definition. The definitions relation contains the same attributes as the PEOPLE relation, and the criteria for each list are the values of the attributes.

3.4.2 Modify-

The "M"odify command takes the user to a criterion screen where the user is asked to submit the name of the mailing list definition to be edited or created. The system confirms creation before actually creating the definition. Of course the user has the option of returning back to the MAIL menu by typing in "QUIT". Once a valid mailing list name is entered into the system, the first definition screen for the mailing list is displayed. This screen is identical
MAIL SYSTEM SCREEN

TYPE THE LETTER OF YOUR CHOICE:

D - DELETE A MAIL LIST AND ITS DEFINITION
L - LIST ALL CURRENT MAILING LISTS
M - CREATE, MODIFY OR BROWSE A DEFINITION
P - PRINT A MAILING LIST, TO STOP PRINTING, ENTER Esc
T - TYPE A MAILING LIST, TO STOP TYPEING, ENTER Esc
Q - QUIT TO MAIN SCREEN
X - EXIT SYSTEM (TYPE "DBASE MAIN" TO RETURN)

Figure 6: Mail Menu
to the person screen. Values are entered into the attributes which the user wants matched for creating the mailing list. All the values on one screen are "and"ed together. All screens of a definition are "or"ed together. That is, any person that matches the entries of any screen in the definition is added to the mailing list. These screens serve as the definition of the mailing list. When the list is created, only those tuples which match this definition are entered into the list.

This retrieval method is based on the "Query By Example" method [DATE82]. The original design was to give the user maximum control in defining the lists, but the use has expanded to general retrieval. The general retrieval is made easy because the user can go from defining a mailing list to typing the list to the screen or printer in one command.

After editing a mail definition screen, the user enters PgDn to display the command menu (figure 7). The commands here are just like the people screen. Added commands are: "A"dd, "D"elete, "M"odify, "P"rint, and "T"ype. The "A"dd and "D"elete commands add and delete the current mail definition screen to or from the current mail list definition. "M"odify returns the user to the mailing list definition criteria screen with the old inputs displayed.
As in the remarks command menu the network view is demonstrated again with the "P"rint and "T"ype commands. These commands are the same respective commands on the Mail Menu (figure 6).

3.4.3 Print-and-Type

The "P"rint and "T"ype commands perform the same operations except at the last step of the function. The "P"rint command routes the output to the printer. The "T"ype command displays the output on the screen.

The user is first prompted for the name of the mailing definition from the criteria screen. The mailing definition is prefilled with the current mailing definition name so the user enters the current name by entering RETURN. The name is then checked for validity and the user is prompted again (with an appropriate message) if the name is not found. If the name is not found the last name entered is displayed so it can be edited, or QUIT can be entered.

Once the mailing list definition is found, the user is then prompted to acknowledge the creation of the list from the definition. If the user responds with a "N"o, the existing list with the current mailing definition name is printed. If "Y" is entered the mailing list is (re)created from the definition. To create a list, each screen of the
MAIL LIST DEFINITION MENU

TYPE THE LETTER OF YOUR CHOICE: 

A - ADD A SCREEN TO DEFINITION
B - MOVE BACKWARDS ONE SCREEN
C - MODIFY OR BROWSE CURRENT SCREEN
D - DELETE CURRENT SCREEN
F - MOVE FORWARD ONE SCREEN
H - HELP ON DEFINING A LIST
P - PRINT A MAILING LIST, TO STOP PRINTING, ENTER Esc
T - TYPE A MAILING LIST, TO STOP TYPING, ENTER Esc
M - CREATE, MODIFY OR BROWSE ANOTHER DEFINITION
Q - QUIT TO MAIN SCREEN
X - EXIT SYSTEM (TYPE "DBASE MAIN" TO RETURN)

Figure 7: Mail Definition Menu
definition is used to find entries in the PERSON relation which match it. If there is a match then the current person is added to the list. After the final definition screen is processed, the duplicate names are removed and the printing begins.
Chapter IV
IMPLEMENTATION AND DESIGN CYCLE

The previous chapter describes the system as it exists presently. This is the result of many redesign cycles with the users. The designer was initially presented with the top level Supervisory Flow Diagram. This left many design decisions to be implemented. One major problem was how to display effectively, the information of a person on one screen. The size limitations of the screen (24 x 80) made it difficult to display the information without documentation. When the users were presented with the initial person information screen, it did not look like the current screen (figure 4).

The attributes for membership class, contact type, and geographical area were all ten character fields. The users wanted these values to be standardized. The best way to implement standardization of these values was to enumerate the possible values and then only allow the users to "mark" or select one. To select a value a mark is made with any character this character is then changed to the character "X" by the system.

After using the system the users found that some of the initial design of the logical sequence of the screens was
not appropriate in some cases. Also some new commands were desired. For example, when looking at a person the users found that it was desirable to determine quickly database entries working for the same company. A similar situation occurred in the mail system, where the users wanted to know the names of the current mailing list definitions. When the users were in the remarks section of the system, they wanted the ability to go directly to the person criterion screen. These requested changes were made by the users through a mail message, a written note, or a new SFD describing the change.

In retrospect the system was developed with one major design phase and then many minor phases. Some of the users' requests were only minor changes while others were major changes in the displayed screens or database. The system was tested on novice users and each test resulted in additional changes. The seemingly endless user-desired modifications might have indicated a bad initial design, but it must be remembered that the system was to become self-documenting. Based on a small group of novice users, the system now has an interface with which a novice user feels comfortable.
Chapter V
EVALUATION

In evaluating this project there are three major areas for examination:

1. the designer's work,
2. the database system dBase II, and
3. the SUPERMAN methodology.

It is difficult to completely separate these three in the analysis, since each area in some way affects the others.

5.1 DESIGNER'S WORK

One way to look at the quality of the design and the designer is to measure the effectiveness of the system. The system is now in constant use. The heavy use results from the system's ability to satisfy the users' needs.

The system fulfills most of the initial design requirements. The requirements that are stated and not directly fulfilled are either no longer needed or some other function satisfies the users. The added requirements for the system were implemented during the implementation cycle. The most important of these was the absence of a user's guide. The consistency requirement has been fulfilled, but there are some commands which do not remain consistent
throughout the system. For example, the command "p" takes the user to the person criterion screen, but in the mail subsystem the same command also means to print a mailing list. This inconsistency was made because the users would rather have the two commands cognitively direct representations of the functions than to have one of the functions represented by an unrelated character.

The expansion of the system functionality requirement is shown by the quick implementation of new user functions. The question of the expandability of the database remains to be answered but the author believes that the system's relations can be easily expanded to meet most any database need of the CSRC (keeping in mind the limitations of dBase II).

5.2 DBASE-II

The dBase II system was the major obstacle in implementing the system. dBase II has its own language to write programs. The language has some positive features like the close coordination of the database design language and the application language. This close coordination exists because the programming language is nearly an extension of the dBase II command language.
The language does have some negative aspects. Because the language is interpreted there are many limitations. One of the limitations of the dBase II system is that only ten files can be open at one time. The three main types of files used in the CSRC system are: database, index, and program files. The database files contain the actual data from a relation. An index file is created when a relation is indexed. The dBase method of subprograms uses program files. This limitation conflicts with the SUPERMAN methodology because the methodology encourages very modular programming and thus many open program files.

Two other limitations of the language are that there could be only sixteen "open" IF statements (nested IF statements) and only sixteen "open" DO statements (nested DO statements). An "open" statement means that the beginning of the statement has been executed but the end of the statement has not been reached. The DO statements include DO WHILE, DO CASE, and DO <file name>. The last statement is the way dBase II calls external subprograms. These limitations were obviously not impossible to design around but the code for the system is just a little less readable because of this. The need for code which is easy to read is lowered because the SFD's represent the system completely and the code is well documented.
5.3 SUPERMAN

The SUPERMAN methodology was a very valuable tool. The initial system was very easy to modify because of its use. More importantly, the SFD's were an exceptional communication medium. The users could make their needs known using the SFD's and the designer could implement the changes quickly and accurately.
6.1 FUTURE WORK

Any future work to extend or modify the system should be easy because of the complete set of SFD's documenting the existing system. The largest foreseeable problem with future work will be the limitations of the dBase II language. With a menu driven system the nesting of IF and DO statements will get very deep. The problem with the IF statements has been eliminated by the use of the interpretive nature of the language, but the number of open DO's the files will cause major problems if the system grows very complex.

Some specific future work would be to add attributes to the PERSON relation. The attributes added would contain information regarding the types of enclosures to send in mailings, whether the member has been approached for funding, and a special salutation for mailings. More information could be kept with each mailing list such as: the dates and purpose of mailing, and names for the script cover letter file. Some of the hierarchical relationships for the member and contact type attributes in the PERSON relation could be removed, so a person could be a primary
contact and a Steering committee member. The remarks subsystem in the people system could be modified to show the number of pending and previous remarks for the current person.

6.2 **DBASE-III**

The previous are just a few possible minor modifications. A major change may take place with the acquisition of dBase III. This is the newest version of dBase and it removes many of the restrictions which were placed on the application programs by the dBase II language. The problem is that a dBase II system is not fully compatible with dBase III. Reportedly, some of the programs must be changed and all of the relations subjected to an upgrade program.
Chapter VII

REFERENCES


Appendix A

BACKUP

The following utility programs, written in EXEC2 (CMS command language), are for backup of the database programs and data. First a brief explanation of procedures to back up system files from the PC to CMS on the main-frame.

1. Set the current directory on the hard disk. (ex. 'CD C: csrc')

2. Set the path for DOS. (ex. 'PATH C: CSRC;C: ')

3. Start up Y-term. Type 'Y'.

4. Type 'T 2400 half space'. (T 4800 for full screen, but line mode will work faster)

5. Logon to CMS userid.

6. Run dbbackup exec by typing 'DBBACKUP'. (this may take a while, but the system tells you it is working at the bottom of the screen).

7. When the program completes, you can logoff.

8. Press the CNTL and BREAK keys at the same time to return to the PC's operating system (DOS).
&TRACE &1

*****************************************************************************
* DBBACKUP EXEC:
* Transfers files from the pc to the main frame.
* Some script control words are then inserted into the files.
* The file names to transfer are in : &DIR_FN &DIR_FT &DIR_FM,
* this file is.
* THE OUTPUT FILE NAME ARE &FN &FT A.
*
*****************************************************************************

*****************************************************************************
* constants
*****************************************************************************
&DIR_FN = DBBACKUP
&DIR_FT = DIR
&DIR_FM = A

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

&PCINDX = 1
MAKEBUF
    LISTFILE &DIR_FN &DIR_FT &DIR_FM (NOHEADER ALL STACK
    &READ VARS ** ** ** &ENUM_TO_COPY
    &TYPE NUMBER TO COPY IS &ENUM_TO_COPY
DROPBUFF
&LOOP -MAIN_LOOP &ENUM_TO_COPY
    &CALL -TRANS_FILE
-MAIN_LOOP
&EXIT

*****************************************************************************
* subroutine TRANS_FILE:
* reads from the file &DIR_FN &DIR_FT &DIR_FM to get the
* file name to transfer.
* creates the DOS file name and calls PCTRANS EXEC to get
* the file.
*****************************************************************************
-TRANS_FILE
&PART1 = C:
MAKEBUF
    EXECIO 1 DISKR &DIR_FN &DIR_FT &DIR_FM &PCINDX
    &READ VARS &FN &FT
    &PCINDX = &PCINDX + 1
    ERASE &FN &FT A
    &DOSNAME = &CONCAT OF &PART1 &FN . &FT
    &TYPE &DOSNAME
    &IF &FT = prg EXEC PCTRANS UP &DOSNAME &FN &FT A
&IF &FT = dbf EXEC PCTRANS UPB &DOSNAME &FN &FT A
DROPBUF &RETURN
*****************************************************************************
Appendix B

SYSTEM CODE
B.1 BROCOMM

* BROCOMM
* ACCESSES THE REMARKS RELATION. ALLOWING TOTAL ACCESS.
* CALLED BY: MAIN
* CALLS: NONE.
*
SELECT PRIMARY
USE REMARKS INDEX REMARKS
SELECT SECONDARY
USE PEOPLE INDEX PNUMBER
SELECT PRIMARY
STORE ' ' TO TARGET
DO WHILE .NOT. UPTOMAIN
**********************************************************************
* DISPLAY_MENU_AND_ACCEPT_ANS
*
**********************************************************************
ERASE
IF TARGET # STR(PERSON:NBR,4)
    STORE STR(PERSON:NBR,4) TO TARGET
    SELECT SECONDARY
    FIND &TARGET
    SELECT PRIMARY
ENDIF
@ 1, 30 SAY 'REMARKS RELATION'
@ $+2, 10 SAY 'NAME: '
@ $, $+1 SAY TRIM(S_FIRST:NAME) + ' ' + S_LAST:NAME
@ $+1, 10 SAY 'COMPANY: '
@ $, $+1 SAY S_ORG:NAME
@ $+1, 10 SAY 'DATE: '
@ $, $+1 SAY DATE
@ $+2, 0 SAY 'REMARKS: '
@ $+1, 2 SAY TEXT1
@ $+1, 2 SAY TEXT2
@ $+1, 2 SAY TEXT3
@ $+1, 2 SAY TEXT4
@ $+3, 10 SAY 'P - FORWARD ONE REMARK'
@ $+1, 10 SAY 'B - BACK ONE REMARK'
@ $+1, 10 SAY 'M - MODIFY REMARK'
@ $+2, 10 SAY 'Q - QUIT TO MAIN MENU'
@ $+1, 10 SAY 'X - EXIT SYSTEM (TYPE "DATABASE MAIN" TO RETURN)'
STORE ' ' TO ANS
@ $+2, 30 SAY 'ENTER CHOICE' GET ANS PICT '!'
READ
**********************************************************************
IF ANS = 'B'
    IF # = 1
        ERASE
        TEXT
YOU ARE AT THE TOP OF THE RELATION
HIT ANY KEY TO RETURN TO PREVIOUS REMARKS
ENDTEXT
WAIT
ELSE
SKIP -1
ENDIF
ENDIF
IF ANS = 'P'
SKIP
IF EOF
ERASE
TEXT
YOU ARE AT THE BOTTOM OF THE RELATION
HIT ANY KEY TO RETURN TO PREVIOUS REMARKS
ENDTEXT
WAIT
GOTO BOTTOM
ENDIF
ENDIF
IF ANS = 'N'
@ 5,15 GET DATE PICT '99/99/99'
@ 8,2 GET TEXT1
@ $+1,2 GET TEXT2
@ $+1,2 GET TEXT3
@ $+1,2 GET TEXT4
READ
ENDIF
IF ANS = 'O'
STORE T TO UPTOMAIN
ENDIF
IF ANS = 'X'
QUIT
ENDIF
ENDDO
STORE '' TO ANS
RETURN
B. 2  DISPERSO

* DISPERSO.PRG
* DISPLAYS THE VALUES FOR THE CURRENT TUPLE IN PEOPLE OR MAIL
* CALLED BY:  SLEEPER
* SEEMAIL
* CALLS : PACK - ENCODES THE BIT INFO, (EX. CONTACT TYPE)
* UNPACK - DECODES THE BIT INFO.

DO UNPACK
ERASE
@ 1,0 SAY 'PERSON SCREEN: ENTER OR MODIFY DATA, '
@ 2,0 SAY 'PERSON NUMBER = '
@ $, $+1 SAY PERSON:NBR
@ $+1,0 SAY 'TITLE' USING 'XXXXXXXXXXXX' GET TITLE PICT 'XXXXX'
@ $,30 SAY 'NICKNAME' USING 'XXXXXXXXXXXX' GET NICK:NAME PICT
@ $+1,0 SAY 'FIRST NAME' USING 'XXXXXXXXXXXX';
  GET FIRST:NAME PICT 'XXXXXXXXXXXXXXX'
@ $,30 SAY 'MID NAME' USING 'XXXXXXXXXXXX' GET MIDD:NAME PICT
@ $+1,0 SAY 'LAST NAME' USING 'XXXXXXXXXXXX';
  GET LAST:NAME PICT 'XXXXXXXXXXXXXXX'
@ $,30 SAY 'SUFFIX' USING 'XXXXXXXXXXXX' GET SUFFIX PICT 'XXXXX'
@ $+1,0 SAY 'PRONOUNCE' USING 'XXXXXXXXXXXX';
  GET PRONOUNCE PICT 'XXXXXXXXXXXXXXX'
@ $,55 SAY 'COMMENTS REMARKS ETC.' USING 'XXXXXXXXXXXXXXXXX'
  GET COMMENTS PICT '
@ $+2,0 SAY 'JOB TITLE' USING 'XXXXXXXXXXXXXXX'
@ $,16 GET JOB:TITLE PICT 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
@ $+1,0 SAY 'COMPANY NAME' USING 'XXXXXXXXXXXXXXX'
@ $,16 GET ORG:NAME PICT 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
@ $, $+1 GET CHECK:ORG PICT 'X'
@ $+1,0 SAY 'STREET'
@ $,16 GET MAIL3 PICT 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
@ $+1,0 SAY 'CITY, STATE, ZIP' USING 'XXXXXXXXXXXXXXX'
@ $,16 GET CITY PICT 'XXXXXXXXXXXXXXXXXXXXX'
@ $, $ GET STATE PICT 'XX'
@ $, $ GET ZIP PICT '99999'
@ $, $ GET ZIP2 PICT '9999'
@ $+2,0 SAY 'CATEGORY' USING 'XXXXXXXXXXXXXXX';
  GET CATEGORY PICT 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
@ $+1,0 SAY 'WORK PHONE' USING 'XXXXXXXXXXXXXXXXXXX';
  GET WORK:PHONE PICT '(999) 999-9999-(9999)'
@ $+1,0 SAY 'MEMBERSHIP DATE' USING 'XXXXXXXXXXXXXXX';
  GET MEMB:DATE PICT '99/99/99'
@ $,46 SAY 'SOURCE NAME' GET SOUR:NAME PICT 'XXXXXXXXXXXXXXX'
@ $+1,0 SAY '
@ $+1,0 SAY 'ORG. MEMB. CLASS'
@ $,30 SAY 'TYPE OF CONTACT'
@ $,60 SAY 'GEOG. AREA'
@ $+1,0 SAY '----------------------'
@ $,30  SAY '------------------'
@ $,56  SAY '------------------'
@ 19,0  SAY 'INDIVIDUAL' USING 'XXXXXXXXXXXXXX' GET PB24 PICT
@ 20,0  SAY 'REGULAR' USING 'XXXXXXXXXXXXXX' GET PB14 PICT
@ 21,0  SAY 'CONTRIBUTOR' USING 'XXXXXXXXXXXXXX' GET PB15 PICT
@ 22,0  SAY 'NON-MEMBER' USING 'XXXXXXXXXXXXXX' GET PB16 PICT
@ 19,30 SAY 'PRIMARY CONTACT' USING 'XXXXXXXXXXXXXX' GET PB19
@ 20,30 SAY 'SECOND CONTACT' USING 'XXXXXXXXXXXXXX' GET PB20
@ 21,30 SAY 'STEERING COMM.' USING 'XXXXXXXXXXXXXX' GET PB21
@ 19,56 SAY 'TECH' USING 'XXXXXX' GET PB11 PICT 'X'
@ 20,56 SAY 'LOCAL' USING 'XXXXX' GET PB12 PICT 'X'
@ 21,56 SAY 'NO VA' USING 'XXXXXX' GET PB13 PICT 'X'
@ 19,68 SAY 'OTHER VA' USING 'XXXXXXXX' GET PB32 PICT 'X'
@ 20,68 SAY 'OUT OF VA' USING 'XXXXXXXX' GET PB33 PICT 'X'
READ
DO PACK
RETURN
GETLIST

GETLIST.PRG
APPENDS THE QUALIFYING PEOPLE TO THE MAIL LIST
CALLED BY: MAKELIST
CALLS: REDUP - REMOVE DUPLICATE PERSON:NBR'S FROM LIST.

THIS PRG IS CALLED WHEN THE PMAIL RELATION IS POINTING TO
NEXT TUPLE TO BE USED IN THE CREATION OF A MAIL LIST.

THE PROGRAM FIRST CREATES FOUR TEST STRINGS. THESE STRINGS
THE MATCHING CRITERIA. THEN EACH STRING "IF'ED" BY USING
THE & (MACRO) SUBSTITUTION.

THE STRINGS ARE CREATED IN ORDER OF THE ATTRIBUTES DEFINED
PERSON RELATION.

ERASE
@ 22.10 SAY 'CREATING LIST FROM DEFINITION .....'
STORE F TO CREATELIST
****************************************************************
* * CREATE THE FIRST TEST STRING
****************************************************************
STORE TRIM(TITLE) TO TEMP
IF TEMP ≠ "'
    STORE TSTRING1 + "TITLE = '10' .AND. " TO TSTRING1
ENDIF
STORE TRIM(FIRST:NAM€) TO TEMP
IF TEMP ≠ "'
    STORE TSTRING1 + "FIRST:NAM€ = '10' .AND. " TO TSTRING1
ENDIF
STORE TRIM(LAST:NAM€) TO TEMP
IF TEMP ≠ "'
    STORE TSTRING1 + "LAST:NAM€ = '10' .AND. " TO TSTRING1
ENDIF
STORE TRIM(PRONOUNCE) TO TEMP
IF TEMP ≠ "'
    STORE TSTRING1 + "PRONOUNCE = '10' .AND. " TO TSTRING1
ENDIF
STORE TRIM(NICK:NAM€) TO TEMP
IF TEMP ≠ "'
    STORE TSTRING1 + "NICK:NAM€ = '10' .AND. " TO TSTRING1
ENDIF
****************************************************************
* * CREATE THE SECOND TEST STRING
****************************************************************

*  *********************************************************************
STORE ' ' TO TSTRING2
STORE TRIM(MIDD:NAME) TO TEMP
IF TEMP = ' ' 
  STORE TSTRING2 + "MIDD:NAME = '10'.AND. " TO TSTRING2
ENDIF
STORE TRIM(SUFFIX) TO TEMP
IF TEMP = ' ' 
  STORE TSTRING2 + "SUFFIX = '10'.AND. " TO TSTRING2
ENDIF
STORE TRIM(CATEGORY) TO TEMP
IF TEMP = ' ' 
  STORE TSTRING2 + "CATEGORY = '10'.AND. " TO TSTRING2
ENDIF
STORE TRIM(JOB:TITLE) TO TEMP
IF TEMP = ' ' 
  STORE TSTRING2 + "JOB:TITLE = '10'.AND. " TO TSTRING2
ENDIF
STORE TRIM(ORG:NAME) TO TEMP
IF TEMP = ' ' 
  STORE TSTRING2 + "ORG:NAME = '10'.AND. " TO TSTRING2
ENDIF
**********************************************************************
*
* CREATE THE THIRD TEST STRING
*
**********************************************************************
STORE ' ' TO TSTRING3
STORE TRIM(MAIL3) TO TEMP
IF TEMP = ' ' 
  STORE TSTRING3 + "MAIL3 = '10'.AND. " TO TSTRING3
ENDIF
STORE TRIM(CITY) TO TEMP
IF TEMP = ' ' 
  STORE TSTRING3 + "CITY = '10'.AND. " TO TSTRING3
ENDIF
STORE TRIM(STATE) TO TEMP
IF TEMP = ' ' 
  STORE TSTRING3 + "STATE = '10'.AND. " TO TSTRING3
ENDIF
STORE TRIM(ZIP) TO TEMP
IF TEMP # ' ' 
  STORE TSTRING3 + "ZIP = '10'.AND. " TO TSTRING3
ENDIF
STORE TRIM(ZIP2) TO TEMP
IF TEMP # ' ' 
  STORE TSTRING3 + "ZIP2 = '10'.AND. " TO TSTRING3
ENDIF
**********************************************************************
* CREATE THE FOURTH TEST STRING
*
*****************************************************************
STORE ' ' TO TSTRING4
IF MEMB:DATE # ' / / ' 
  STORE TSTRING4 + "MEMB:DATE = '&MEMB:DATE'.AND." TO TSTRING4
ENDIF
STORE TRIM(SOUR:NAME) TO TEMP
IF TEMP # ' 
  STORE TSTRING4 + "SOUR:NAME = '10'.AND." TO TSTRING4
ENDIF
STORE GEOG:AREA TO TEMP
IF TEMP # ' 
  STORE TSTRING4 + "GEOG:AREA = '10'.AND." TO TSTRING4
ENDIF
STORE CONT:TYPE TO TEMP
IF TEMP # ' 
  STORE TSTRING4 + "CONT:TYPE = '10'.AND." TO TSTRING4
ENDIF
STORE MEMB:TYPE TO TEMP
IF TEMP # ' 
  STORE TSTRING4 + "MEMB:TYPE = '10'.AND." TO TSTRING4
ENDIF
IF COMMENTS # ' 
  STORE TSTRING4 + "COMMENTS # " TO TSTRING4
ENDIF
IF CHECK:ORG # ' 
  STORE TSTRING4 + "CHECK:ORG # " TO TSTRING4
ENDIF
*****************************************************************
* LOOP 4 TIMES TO REMOVE THE EXTRA .AND. FROM THE END OF STRING
* IF THE STRING IS JUST 1 IN LENGTH STORE 'T' = 'T' TO THE STRING
*
STORE 1 TO INDX
DO WHILE INDX < 5
  STORE STR(INDX,1) TO CINDX
  STORE 'TSTRING' + CINDX TO TSTRING
  STORE 'LENGTH' + CINDX TO LENGTH
  STORE LEN(&TSTRING) TO &LENGTH
  IF &LENGTH # 1
    STORE &LENGTH - 7 TO &LENGTH
    STORE $(&TSTRING,1,&LENGTH) TO &TSTRING
  ELSE
    STORE "'T' = 'T'" TO &TSTRING
  ENDIF
  STORE INDX + 1 TO INDX
ENDDO
IF LENGTH1 = 1 .AND. LENGTH2 = 1 .AND. LENGTH3 = 1 .AND. ;
    LENGTH4 = 1
    ERASE
    STORE ' ' TO YESNO
    @ 6,10 SAY 'THERE IS A BLANK SCREEN IN THIS DEFINITION'
    @ 8,10 SAY 'THIS WILL CAUSE ALL CURRENT MEMBERS IN THE'
    @ 10,10 SAY 'DATABASE TO BE INCLUDED IN THE LIST'
    @ 12,10 SAY 'DO YOU WISH TO CONTINUE TYPE "Y" FOR YES,'
    @ 14,10 SAY 'ANYTHING ELSE WILL TAKE YOU BACK TO THE MAIN'
    @ 16,10 SAY 'MAIL MENU'
    @ 18,10 SAY 'ENTER "Y" OR "N" GET YESNO PICT ’!'
    READ
    IF YESNO = 'Y'
        STORE T TO CREATELIST
    ENDIF
ELSE
    STORE T TO CREATELIST
ENDIF
IF CREATELIST
    SELECT PRIMARY
    USE PEOPLE
    SELECT SECONDARY
    USE MTEMP
    SELECT PRIMARY
ENDIF

********************************************************************************************************************
*
* LOOP FOR ALL TUPLES IN THE PEOPLE RELATION AND SEE IF TUPLE
* PRODUCES A TRUE FOR ALL FOUR TEST STRINGS. IF SO THEN COPY
* ADDRESS INFO INTO THE TMAIL RELATION.
*
********************************************************************************************************************
DO WHILE .NOT. EOF
    IF &TSTRING1
    IF &TSTRING2
    IF &TSTRING3
    IF &TSTRING4
        SELECT SECONDARY
        APPEND BLANK
        REPLACE S:PERSON:NBR WITH P:PERSON:NBR
        REPLACE ZIP:PLUS WITH ZIP + ZIP2
        REPLACE LINEONE WITH TRIM(TITLE) + TRIM(FIRST:NAME)
        + TRIM(MIDD:NAME) + TRIM(LAST:NAME) + SUFFIX
        REPLACE LINETWO WITH JOB:TITLE
        REPLACE LINETHREE WITH ORG:NAME
        REPLACE LINEFOUR WITH MAIL3
        STORE TRIM(CITY) + ' ' + STATE + ' ' + ZIP + ;
        ' ' + ZIP2 TO TEMP
        REPLACE LINFIVE WITH TEMP
        SELECT PRIMARY
    ENDIF
ENDIF
ENDIF
ENDIF
SKIP
ENDDO.
USE
SELECT SECONDARY
APPEND BLANK
REPLACE ZIP:PLUS WITH '999999999'
USE
SELECT PRIMARY
USE MTEMP
INDEX ON STR(PERSON:NBR,4) TO MTEMP
DO REDUP
USE
ENDIF
RETURN
B.4 GETMNAME

* GETMNAME
* PROMPTS THE USER TO GIVE A NAME OF A MAIL DEFINITION
* CALLED BY: MAKELIST
* CALLS : SEEMAIL
* IF NAMEFOUND IS SET TO FALSE THE NAME NOT FOUND
* MESSAGE WILL BE DISPLAYED.
*
STORE MAIL:NAME + ' ' TO MAIL:NAME
STORE $MAIL:NAME,1,10 TO MAIL:NAME
USE PMAIL INDEX PMAIL
ERASE
IF NOT NAMEFOUND
  @ 4,10 SAY 'THE NAME ENTERED WAS NOT FOUND'
ENDIF
@ 7,10 SAY 'ENTER MAIL LIST NAME OR "QUIT" TO RETURN TO MENU'
@ 10,10 SAY 'MAIL LIST NAME ' GET MAIL:NAME PICT '!!!!!!!!!!!!'
READ
STORE TRIM(MAIL:NAME) TO MAIL:NAME
IF (MAIL:NAME = 'QUIT')
  STORE T TO UPTOMAIL
ENDIF
RETURN
B.5 GETNAME

* GETNAME
* TO GET THE NAME AND/OR ORGANIZATION OF A PERSON.
* CALLED BY: PEOPLE
* CALLS : NONE.

GET_Criteria

* ERASE

TEXT
ENTER ONE OR MORE CRITERIA FOR IDENTIFYING PERSON
PGDN OR <RETURN> PAST LAST FIELD TO PROCEED
A NEW PERSON WILL BE CREATED ONLY WHEN YOU ENTER
INFORMATION IN THE "LAST NAME" FIELD (OR THE "LAST NAME
AND "FIRST NAME" FIELDS), AND A PERSON WITH THIS
NAME IS NOT FOUND AMONG THE CURRENTLY EXISTING PERSONS
ENDTEXT
@ 11,10 SAY 'LAST NAME' USING 'XXXXXXXXXX';
   GET MEM:TLN PICT 'XXXXXXXXXXXXXXX';
@ $+1,10 SAY 'FIRST NAME' USING 'XXXXXXXXXX';
   GET MEM:TFN PICT 'XXXXXXXXXXXXXXX';
@ $+1,10 SAY 'ORGANIZATION NAME' GET MEM:TORG;
   PICT 'XXXXXXXXXXXXXXXXXXXXXXXXX';
@ $+4,10 SAY 'Q - QUIT AND RETURN TO MAIN MENU'
@ $+1,10 SAY 'X - EXIT FROM SYSTEM'
STORE ' ' TO ANS
@ $+2,20 GET ANS PICT '!' READ
RETURN
B.6  LABELS

*  LABELS
  *  MAKES AND PRINTS THE MAILLING LIST EITHER TO SCREEN
  *  OR PRINTER
  *  CALLED BY:  MAIL
  *  CALLS  :  MAKELIST
  *
  ERASE
  DO  MAKE LIST
  IF  UPTOMAIL
     RETURN
  ENDIP
  STORE  "" ;
  TO  FMT
  STORE  'USING ' +  FMT  TO  FMT
  SET  FORMAT  TO  &PRINTFMT
  ERASE
  USE  &MAIL:NAME
  STORE  0  TO  TOPLABEL
  DO  WHILE  .NOT.  EOF
     STORE  LIN ONE  TO  L1
     STORE  LIN T W O  TO  L2
     STORE  LINETHREE  TO  L3
     STORE  LINEFOUR  TO  L4
     STORE  LINEFIVE  TO  L5
     SKIP  1
     @  TOPLABEL,0  SAY  L1 + ' ' + LIN ONE  &FMT
     @  TOPLABEL + 1,0  SAY  L2 + ' ' + LINET W O  &FMT
     @  TOPLABEL + 2,0  SAY  L3 + ' ' + LINETHREE  &FMT
     @  TOPLABEL + 3,0  SAY  L4 + ' ' + LINEFOUR  &FMT
     @  TOPLABEL + 4,0  SAY  L5 + ' ' + LINEFIVE  &FMT
     SKIP  1
     STORE  TOPLABEL  +  6  TO  TOPLABEL
  IF  PRINTFMT  =  'SCREEN'  .AND.  TOPLABEL  >  21  
     WAIT
     ERASE
     STORE  0  TO  TOPLABEL
  ELSE
     IF  TOPLABEL  >  61
        STORE  0  TO  TOPLABEL
        ENDIF
     ENDIF
  ENDIF
  ENDDO
  IF  PRINTFMT  =  'SCREEN'
     WAIT
  ELSE
     @  1,1  SAY  ''
  ENDIF
  SET  FORMAT  TO  SCREEN
STORE '' TO ANS
RETURN
B.7  MAIL

* MAIL
* DISPLAYS THE MENU AND EXECUTES THE COMMANDS.
* CALLED BY: MAIN
* CALLS : LABELS
*            SEEMAIL
*            
* AS IN THE MAIN PROGRAM THE VAR COMMAND CONTAINS 'DO <FN>'.
* IF THE COMMAND SELECTED DOES NOT NEED TO 'DO' ANOTHER FILE
* THE STATEMENT &COMMAND EXECUTES NOTHING WHICH IS JUST A
* RETURN STATEMENT.
*
STORE F TO GOTNAME
STORE ' ' TO MAIL::NAME
DO WHILE .NOT. UPTOMAIN
STORE F TO UPTOMAIL
IF ANS = 'PRINT' .OR. ANS = 'TYPE' .OR. ANS = 'MODIFY'
    STORE $(ANS,1,1) TO ANS
ELSE

******************************************************************************
*
* DISPLAY_MENU_AND_ACCEPT_ANS
*
******************************************************************************

STORE ' ' TO ANS
ERASE
@ 2,10 SAY 'MAIL SYSTEM SCREEN'
@ 3,10 SAY 'ENTER THE LETTER OF YOUR CHOICE' GET ANS
@ 5,10 SAY 'D - DELETE A MAIL LIST AND ITS DEFINITION'
@ $+1,10 SAY 'L - LIST ALL CURRENT MAILING LISTS'
@ $+1,10 SAY 'M - CREATE, MODIFY OR BROWSE A DEFINITION'
@ $+1,10 SAY 'P - PRINT A MAILING LIST, TO STOP ENTER ESC'
@ $+1,10 SAY 'T - TYPE A MAILING LIST, TO STOP ENTER ESC'
@ $+2,10 SAY 'Q - QUIT TO MAIN SCREEN'
@ $+1,10 SAY 'X - EXIT SYSTEM (TYPE "DBASE MAIN" TO RETURN)'
READ
ERASE
******************************************************************************
ENDIF
STORE 'DO NOTHING' TO COMMAND
IF 'D' = ANS
******************************************************************************
*
* DELETE_MAIL_LIST
*
******************************************************************************

STORE T TO NAMEFOUND
STORE F TO GOTNAME
DO WHILE .NOT. GOTNAME .AND. .NOT. UPTOMAIL
    DO GETNAME
SELECT PRIMARY
USE PMAIL INDEX PMAIL
STORE TRIM(MAIL:NAMEl) TO TARGET
FIND &TARGET
IF # = 0
STORE P TO NAMEFOUND
ELSE
STORE T TO GOTNAME
ERASE
STORE ' ' TO YESNO
@ 10,10 SAY 'ARE YOU SURE?';
GET YESNO PICT '!
READ
ERASE
IF YESNO = 'Y'
IF FILE(MAIL:NAME)
DELETE FILE &MAIL:NAME
ENDIF
DO WHILE MAIL:LIST = MAIL:NAMEl AND NOT. EOF
DELETE
SKIP
ENDDO
PACK
USE
@ 10,10 SAY 'FILE AND DEFINITION DELETED'
ELSE
@ 10,10 SAY 'FILE AND DEFINITION NOT DELETED'
ENDIF
WAIT
ENDIF
ENDDO
ENDIF
***********************************************************************
IF 'L' = ANS
***********************************************************************
*
LIST_ALL_MAILLIST_DEFINITIONS  *
*
***********************************************************************
@ 2,19 SAY 'MAIL LIST NAME    ADDRESSES'
@ 3,19 SAY '----------    ---------'
@ 4,19 SAY ' '
SELECT PRIMARY
USE PMAIL INDEX PMAIL
DO WHILE NOT. EOF
STORE MAIL:LIST TO OLDDEF
IF FILE(OLDDEF)
SELECT SECONDARY
USE &OLDDEF
GOTO BOTTOM
STORE # - 1 TO NUMBER
USE
SELECT PRIMARY
ELSE
STORE 0 TO NUMBER
ENDIF
@ $+1, 20 SAY OLDDEF
@ $+1, 36 SAY NUMBER
DO WHILE MAIL:LIST = OLDDEF .AND. .NOT. EOF
  SKIP
ENDDO
ENDIF
IF MAIL:LIST # OLDDEF
  @ $+1, 20 SAY MAIL:LIST
ENDIF
USE
WAIT
ENDIF
****************************************************************
IF 'M' = ANS
  STORE 'DO SEEMAIL' TO COMMAND
* THIS PROC RETURNS WITH A NEW VALUE FOR ANS THE USER CAN GO
* INTO THE PRINTING OF A MAILIST.
ENDIF
IF 'P' = ANS
  STORE 'PRINT' TO PRINTFMT
  STORE 'DO LABELS' TO COMMAND
ENDIF
IF 'T' = ANS
  STORE *SCREEN* TO PRINTFMT
  STORE 'DO LABELS' TO COMMAND
ENDIF
IF 'Q' = ANS
  STORE T TO UPTOMAIN
ENDIF
IF 'X' = ANS
  QUIT
ENDIF
&COMMAND
ENDDO
STORE '' TO ANS
RETURN
B.3 MAIN

* MAIN.PRG
* DISPLAYS THE MAIN MENU
* CALLED BY: NONE.
* CASSL : BROCOMM - ACCESS THE REMARKS RELATION.
  PEOPLE - ACCESS THE PEOPLE RELATION.
* MAIL - ENTER THE MAIL SYSTEM.
* IN ORDER TO SAVE AN "OPEN" IF STATEMENT THE PRG STORES
* 'DO <FN> TO COMMAND. AT THE BOTTOM OF THE MAIN LOOP THE VAR
* COMMAND IS EXECUTED BY THE STATEMENT &COMMAND. IF NO COMMAND
* NEEDS TO BE EXECUTED THE COMMAND CALLS A FILE NAMED NOTHING.
* 
SET BELL OFF
SET CONSOLE ON
SET DEBUG OFF
SET DELETE OFF
SET ECHO OFF
SET STEP OFF
SET TALK OFF
SET FORMAT TO SCREEN
STORE F TO QUIT
DO WHILE .NOT. QUIT
  *
  DISPLAY_MAIN_MENU_AND_ACCEPT_ANS
  *
USE
* THIS CLEAR IS TO KEEP MEMORY CLEAN
CLEAR
STORE F TO QUIT
STORE F TO UPTOMAIN
ERASE
STORE ' ' TO ANS
@ 3,10 SAY 'C S R C D A T A B A S E M A I N M E N U'
@ 4,10 SAY '---------- ---------- ----------'
@ 6,10 SAY 'TYPE THE LETTER OF YOUR CHOICE' GET ANS PICT '!'
@ 8,10 SAY 'P - ENTER, MODIFY, OR SEE INFORMATION FOR A PERSON'
@ $+1,10 SAY 'M - PRODUCE MAILING LABELS'
@ $+1,10 SAY 'R - BROWSE REMARKS, PHONE MESSAGES, ETC.'
@ $+2,10 SAY 'Q - EXIT CSRC SYSTEM (TYPE "DO MAIN" TO RETURN)'
@ $+1,10 SAY 'X - EXIT SYSTEM (TYPE "DBASE MAIN" TO RETURN)'
READ
STORE 'DO NOTHING' TO COMMAND
IF ANS = 'P'
  STORE 'DO PEOPLE' TO COMMAND
ENDIF
IF ANS = 'M'
  STORE 'DO MAIL' TO COMMAND
ENDIF
IF ANS = 'R'
    STORE 'DO BROCOMM' TO COMMAND
ENDIF
IF ANS = 'Q'
    STORE T TO QUIT
ENDIF
IF ANS = 'X'
    QUIT
ENDIF
&COMMAND
ENDDO
ERASE
B.9  **MAKELIST**

* **MAKELIST**
* **CALLED BY:** LABELS
* **CALLS:** GETLIST - ADDS THE CURRENT TOUPLE IN PEOPLE
  RELATION TO THE MTEMP RELATION.
* **GETMNAME** - GETS A MAIL LIST NAME FROM THE USER.
* **THIS PROGRAM USES THE PMAIL RELATION TO CREATE THE MAIL LIST**
* THE PMAIL RELATION DEFINES MANY MAIL LISTS. AFTER THE USER
  APPROVES THE (RE)CREATION OF THE MAIL LIST, THIS PRG FINDS
  BEGGING OF THE WANTED DEFINITION AND LOOPS THROUGH EACH
  CALLING GETLIST TO APPEND ANY PERSON WHICH IS RETRIEVED
  USING THE CURRENT TOUPLE.
* **STORE T TO NAMEFOUND**
**STORE F TO GOTNAME**
**DO WHILE .NOT. UPTOMAIL .AND. .NOT. GOTNAME**
  **DO GETMNAME**
  **USE PMAIL INDEX PMAIL**
  **FIND &MAIL:NAME**
  **IF # <> 0**
    **STORE T TO GOTNAME**
    **STORE MAIL:LIST TO MAIL:NAME**
  **ELSE**
    **STORE F TO NAMEFOUND**
  **ENDIF**
**ENDDO**
**IF UPTOMAIL**
  **RETURN**
**ENDIF**
**STORE F TO GOTYESNO**
**STORE ' ' TO YESNO**
**DO WHILE .NOT. GOTYESNO**
  **ERASE**
  @ 9.10 SAY 'THE DEFINITION HAS BEEN FOUND, IF YOU WANT THE'  
  @ 10.10 SAY 'LIST TO BE CREATED (DESTROY OLD LIST),'  
    ENTER "Y" FOR YES'
  @ 11.10 SAY 'ENTER "N" TO USE THE EXISTING LIST';  
    GET YESNO PICT '!'
  **READ**
  **IF YESNO = 'Y' .OR. YESNO = 'N'**
    **STORE T TO GOTYESNO**
  **ENDIF**
**ENDDO**
**IF YESNO = 'N'**
  **RETURN**
**ENDIF**
**DELETE FILE MTEMP**
**USE MAILFORM**
COPY TO MTEMP
USE
SELECT PRIMARY
USE PMAIL INDEX PMAIL
FIND &MAIL:NAME
STORE # TO RECNUN
DO WHILE (RECNUN # 0)
  DO GETLIST
  SELECT PRIMARY
  USE PMAIL INDEX PMAIL
  GOTO RECNUN
  SKIP
  STORE # TO RECNUN
  IF (MAIL:LIST # MAIL:NAME .OR. EOF .OR. .NOT. CREATELIST)
    STORE 0 TO RECNUN
  ENDIF
ENDDO
IF CREATELIST
  USE MTEMP
  SORT ON ZIP:PLUS TO &MAIL:NAME
ENDIF
SELECT SECONDARY
USE
SELECT PRIMARY
USE
RETURN
B.10  NOTHING

*  NOTHING
*  DOES NOTHING
*  CALLED BY: GETNAME
*  MAIN
*  MAIL
*  PEOPLE
*  CALLS   : NONE.
RETURN
B.11 PACK

* PACK
* ENCODE THE VARS FROM THE SCREEN TO THE RELATION TUPLE
* CALLED BY: DISPERSION
* CALLS : NONE.
*
STORE ' ' TO PEOG:AREA
STORE ' ' TO PONT:TYPE
STORE ' ' TO PEMB:TYPE
IF PB33 # ' '
    STORE 'O' TO PEOG:AREA
ENDIF
IF PB32 # ' '
    STORE 'Y' TO PEOG:AREA
ENDIF
IF PB13 # ' '
    STORE 'N' TO PEOG:AREA
ENDIF
IF PB12 # ' '
    STORE 'L' TO PEOG:AREA
ENDIF
IF PB11 # ' '
    STORE 'T' TO PEOG:AREA
ENDIF
IF PB20 # ' '
    STORE 'S' TO PONT:TYPE
ENDIF
IF PB19 # ' '
    STORE 'P' TO PONT:TYPE
ENDIF
IF PB21 # ' '
    STORE 'C' TO PONT:TYPE
ENDIF
IF PB24 # ' '
    STORE 'I' TO PEMB:TYPE
ENDIF
IF PB14 # ' '
    STORE 'R' TO PEMB:TYPE
ENDIF
IF PB15 # ' '
    STORE 'C' TO PEMB:TYPE
ENDIF
IF PB16 # ' '
    STORE 'N' TO PEMB:TYPE
ENDIF
IF MEMB:TYPE # PEMB:TYPE
    REPLACE MEMB:TYPE WITH PEMB:TYPE
ENDIF
IF CONT:TYPE # PONT:TYPE
REPLACE CONT::TYPE WITH PONT::TYPE
ENDIF
IF GEOG::AREA # PEOG::AREA
    REPLACE GEOG::AREA WITH PEOG::AREA
ENDIF
RETURN
B.12  PEOPLE

* PEOPLE
* TO GET THE NAME AND/OR ORGANIZATION OF A PERSON.
* CALLED BY: MAIN
* CALLS : GETNAME - PROMPTS THE USER FOR SEARCH CRITERIA
* : SEEPERSON - MANAGES THE PERSON SCREEN.
* : NOTHING

* GETS INFORMATION FOR THE RETRIEVAL THEN BASED ON INPUT
* THE RELATION IS INDEXED. IF THE PERSON NAME WAS GIVEN
* AND THE PERSON IS NOT FOUND, CONFIRMATION IS REQUESTED.
* LIKE THE MAIN PRG THE VAR COMMAND CONTAINS THE STRING
* "DO <FN>"
* THIS IS EXECUTED OUT FROM AS MANY IF STATEMENTS AS POSSIBLE.
* STORE ' ' TO MEM:TLN
STORE ' ' TO MEM:TFN
STORE ' ' TO MEM:TORG
DO WHILE .NOT. UPTOMAIN
*
* GET CRITERIA.
*
DO GETNAME
STORE F TO UPTOPEOPLE
STORE 'DO NOTHING' TO COMMAND
*
* TRIM THE INPUTS
*
STORE MEM:TLN + MEM:TFN TO MEM:LN
STORE TRIM(MEM:LN) TO MEM:LN
STORE TRIM(MEM:TORG) TO MEM:ORG
*
* EVALUATE ONLY THE ANS IF IT IS BLANK.
*
IF ANS = ''
IF ANS = 'X'
QUIT
ENDIF
IF ANS = 'Q'
STORE T TO UPTOMAIN
ENDIF
ELSE
IF MEM:LN = '' .AND. MEM:ORG = ''
******************************************************************************
* FINDPERSON
* ONLY THE PERSON'S NAME WAS GIVEN.
* *
******************************************************************************
USE PEOPLE INDEX PNAME$„PCOMPANY„PNUMBER
STORE 'LAST NAME' TO INDEX1
FIND &MEM:LN
IF # = 0
ERASE
STORE ' ' TO ENTERYN
@ 10, 10 SAY 'A PERSON WAS NOT FOUND WITH THAT NAME'
@ 12, 10 SAY 'WOULD YOU LIKE TO ENTER A PERSON ? (Y/N)'
GET ENTERYN PICT '!
READ
IF ENTERYN = 'Y'
RESTORE FROM PNUMBER ADDITIVE
APPEND BLANK
REPLACE PERSON:NBR WITH PNUMBER
STORE PNUMBER + 1 TO PNUMBER
SAVE ALL LIKE PNUMBER TO PNUMBER
REPLACE LAST:NAME WITH MEM:LN
REPLACE FIRST:NAME WITH MEM:TFN
STORE 'DO SEEPERSON' TO COMMAND
ENDIF
ELSE
STORE 'DO SEEPERSON' TO COMMAND
ENDIF
ENDIF
IF MEM:ORG # ' '
****************************************************************
********** ********** **** ****
** FIND_ORG_AND_PERSON **
* THE PERSON ORGANIZATION WAS GIVEN. *
**
****************************************************************
USE PEOPLE INDEX PCOMPANY„PNAME$„PNUMBER
STORE 'ORGANIZATION' TO INDEX1
FIND &MEM:ORG
* WAS THE ORGANIZATION FOUND ???
* 
IF # = 0
* NO THE ORG WAS NOT FOUND
ERASE
TEXT
A PERSON WITH THIS ORGANIZATION NAME WAS NOT FOUND
HIT ANY KEY TO RETURN TO SELECTION SCREEN
ENDTEXT
WAIT
ELSE
* YES THE ORG WAS FOUND
*
* WAS A NAME GIVEN ALSO?
* IF MEM:LN # ' ' *
* TRY TO FIND THE PERSON'S NAME *
* STORE F TO MATCH
  DO WHILE .NOT. MATCH .AND. ORG:NAME = MEM:ORG ;
    .AND. .NOT. EOF
    IF LAST:NAME + FIRST:NAME = MEM:LN
    STORE T TO MATCH
    ELSE
    SKIP 1
  ENDF
  ENDDO
  IF .NOT. MATCH
  ERASE
  TEXT
A PERSON WITH THIS LAST NAME AND ORGANIZATION NAME WAS NOT FOUND
HIT ANY KEY TO RETURN TO SELECTION SCREEN
ENDDTEXT
WAIT
ELSE
  STORE 'DO SEEPESSON' TO COMMAND
  ENDF
ELSE
* PERSON'S NAME WAS NOT GIVEN *
* STORE 'DO SEEPESSON' TO COMMAND
  ENDF
  ENDF
ENDDIF
ENDDIF
ENDDIF
ENDDIF
****************************************************************
&COMMAND
****************************************************************
&COMMAND
ENDDO
RETURN
B.13  REDUP-

* REDUP
* TO REMOVE DUPLICATES IN THE MAILLING LIST
* CALLED BY: GETLIST
* CALLS : NONE.
* THE RELATION IS INDEX BY PERSON:NBR.

GOTO TOP
STORE PERSON:NBR TO OLDPNUM
SKIP
DO WHILE .NOT. EOF
   IF OLDPNUM = PERSON:NBR
      DELETE
   ELSE
      STORE PERSON:NBR TO OLDPNUM
      ENDDO
   SKIP
ENDDO
PACK
RETURN
B.14 REINDEX

* REINDEX
* RECREATES ALL OF THE RELATIONS INDEX FILES
* CALLED BY: NONE.
* CALLES : NONE

SET TALK ON
USE PEOPLE
INDEX ON LAST:NAME + FIRST:NAME TO PNAME
INDEX ON ORG:NAME TO PCOMPANY
INDEX ON PERSON:NBR TO PNUMBER
USE REMARKS
INDEX ON PERSON:NBR TO REMARKS
USE PMAIL
INDEX ON MAIL:LIST TO PMAIL
SET TALK OFF
RETURN
B.15 SEECOM

* SEECOM
* TO ACCESS THE COMMENTS FOR A PERSON
* CALLED BY: SEEPEPERSON
* CALLS : NONE.
* THE COMMENTS ARE INDEXED BY PERSON:NBR, THE USER CAN ONLY
* ACCESS THE COMMENTS FOR THE CURRENT PERSON.
* BEFORE RETURNING, A CHECK IS MADE TO SEE IF THE USER DELETED
* ALL OF THE COMMENTS. IF SO THE APPROPRIATE FIELD IS MARKED
* IN THE PEOPLE RELATION.
*
SELECT SECONDARY
USE REMARKS INDEX REMARKS
STORE STR(P.PEOPLE:NBR,4) TO TARGET
FIND &TARGET
IF # = 0
    APPEND BLANK
    REPLACE S.PEOPLE:NBR WITH P.PEOPLE:NBR
    REPLACE DATE WITH DATE()
ENDIF
STORE F TO QUIT
DO WHILE .NOT. QUIT
  ERASE
  @ 1,10 SAY 'PERSON NAME =' ' 
  @ 1,$+2 SAY TRIM(P.FIRST:NAME) + ' ' + P.LAST:NAME
  IF *
    @ 1,70 SAY 'DELETED'
  ENDIF
  @ $+2,10 SAY 'ORGANIZATION ='
  @ $+2,2 GET TEXT1
  @ $+1,2 GET TEXT2
  @ $+1,2 GET TEXT3
  @ $+1,2 GET TEXT4
  READ
  @ 12,10 SAY 'D - DELETE CURRENT REMARK'
  @ $+1,10 SAY 'A - ADD A REMARK '
  @ $+1,10 SAY 'F - FORWARD ONE SCREEN FOR THIS PERSON'
  @ $+1,10 SAY 'B - BACK ONE SCREEN OF REMARKS FOR THIS PERSON'
  @ $+1,10 SAY 'U - UNDELETE CURRENT REMARK'
  @ $.+2,10 SAY 'C - RETURN TO PERSON INFORMATION SCREEN'
  @ $+1,10 SAY 'P - ENTER MODIFY OR BROWSE ANOTHER PERSON'
  @ $+1,10 SAY 'Q - QUIT TO MAIN MENU'
  @ $+1,10 SAY 'X - EXIT SYSTEM (TYPE "DBASE MAIN" TO RETURN)'
  STORE ' ' TO ANS
  @ $+2.5 SAY 'ENTER CHOICE' GET ANS PICT '!'
IF ANS = 'D'
DELETE
ENDIF
IF ANS = 'A'
APPEND BLANK
REPLACE S.PESON:NBR WITH P.PESON:NBR
REPLACE DATE WITH DATE()
ENDIF
IF ANS = 'U'
RECALL
ENDIF
IF ANS = 'F'
SKIP 1
IF S.PESON:NBR # P.PESON:NBR .OR. EOF
ERASE
TEXT
NO MORE MESSAGES FOR THIS PERSON
HIT ANY KEY TO RETURN TO PREVIOUS REMARK SCREEN
ENDTEXT
WAIT
SKIP -1
ENDIF
ENDIF
IF ANS = 'B'
SKIP -1
IF S.PESON:NBR # P.PESON:NBR
ERASE
TEXT
NO MORE MESSAGES FOR THIS PERSON
HIT ANY KEY TO RETURN TO PREVIOUS REMARK SCREEN
ENDTEXT
WAIT
SKIP 1
ENDIF
ENDIF
IF 'C' = ANS
STORE T TO QUIT
ENDIF
IF 'P' = ANS
STORE T TO UPTOPEOPLE
STORE T TO QUIT
ENDIF
IF 'Q' = ANS
STORE T TO UPTMAIN
STORE T TO QUIT
ENDIF
IF 'X' = ANS
QUIT
ENDIF
ENDDO
STORE F TO QUIT
STORE ' ' TO Ans
FIND &TARGET
DO WHILE TARGET = STR(S.PE RSON:NBR,4)
   IF TEXT1 = ' ' .AND. TEXT2 = ' ' .AND. ;
      TEXT3 = ' ' .AND. TEXT4 = ' ' ;
      .OR.*
      DELETE
      PACK
      FIND &TARGET
   ELSE
      SKIP 1
   ENDDO
ENDIF
FIND &TARGET
STORE # TO TARGET
USE
SELECT PRIMARY
IF TARGET = 0
   REPLACE COMMENTS WITH ' '
ENDIF
RETURN
SEEMAIL

* THIS PRG FIRST GETS THE NAME OF THE DEFINITION AND
* THEN IT DISPLAYS THE PERSON SCREENS.
* CALLED BY: MAIL       RETURNS: EITHER A CHARACTER IN ANS
* CALLS : GETNAME       OR 'PRINT', 'TYPE' IN ANS
* DISPERSON

********************************************************************
* GET_VALID_MAIL_LIST_NAME
* ********************************************************************
STORE T TO NAMEFOUND
STORE F TO GOTNAME
DO WHILE .NOT. GOTNAME .AND. .NOT. UPTOMAIL
  DO GETNAME
    IF .NOT. UPTOMAIL
      USE PMAIL INDEX PMAIL
      STORE TRIM(MAIL:NAME) TO TARGET
      FIND &TARGET
      IF # <> 0
        STORE T TO GOTNAME
      ELSE
        ERASE
        STORE ' ' TO YESNO
        @ 10,10 SAY 'THE MAIL LIST DEFINITION '
        @ 10,$+1 SAY MAIL:NAME
        @ 12,10 SAY 'WAS NOT FOUND, WOULD YOU LIKE TO CREATE IT?' ;
        GET YESNO PICT 'Y'
        READ
        IF YESNO = 'Y'
        STORE T TO GOTNAME
        ELSE
        STORE F TO NAMEFOUND
      ENDDIF
    ENDDIF
  ENDDO
  IF UPTOMAIL
    RETURN
  ENDFILE
  FIND &MAIL:NAME
  STORE # TO TEMP
  IF TEMP = 0
    APPEND BLANK
    REPLACE MAIL:LIST WITH MAIL:NAME
  ENDFILE
STORE MAIL:LIST TO MAIL:NAME
***************************************************************************
*  *  DISPLAY_INSTRUCTIONS
* ***************************************************************************
ERASE
STORE " " TO ANS
DO WHILE .NOT. UPTOMAIL .AND. .NOT. UPTOMAIN
   IF 'H' # ANS .AND. " " # ANS
       DO DISPERSON
   ENDIF
***************************************************************************
*  *  DISPLAY_MENU_AND_ACCEPT_ANS
* ***************************************************************************
ERASE
TEXT

MAIL LIST DEFINITION MENU

A - ADD A SCREEN TO DEFINITION
B - MOVE BACKWARDS ONE SCREEN
C - MODIFY OR BROWSE CURRENT SCREEN
D - DELETE CURRENT SCREEN
F - MOVE FORWARD ONE SCREEN
H - HELP ON DEFINING A LIST
M - EDIT, MODIFY, OR BROWSE ANOTHER LIST'S DEFINITION
P - PRINT A MAILING LIST, TO STOP PRINTING, ENTER ESC
T - TYPE A MAILING LIST, TO STOP TYPING, ENTER ESC
Q - QUIT TO MAIN MENU
X - EXIT SYSTEM (TYPE "DBASE MAIN" TO RETURN)

ENDTEXT
STORE " " TO ANS
@ 2,10 SAY 'CURRENT MAIL DEFINITION IS: '
@ 2,$+1 SAY MAIL:NAME
@ 5,10 SAY 'TYPE THE LETTER OF YOUR CHOICE' GET ANS PICT '!!'
READ
***************************************************************************
ERASE
IF 'A' = ANS
   APPEND BLANK
   REPLACE MAIL:LIST WITH MAIL:NAME
   STORE # TO TEMP
   USE
   USE PMAIL INDEX PMAIL
GOTO TEMP
ENDIF
IF 'B' = ANS
STORE # TO TEMP
SKIP -1
STORE # TO TTEMP
IF MAIL:LIST = MAIL:NAME OR. TTEMP = TEMP
  TEXT
  YOU ARE AT THE FIRST RECORD OF THE MAIL LIST DEFINITION.
  ENDTEXT
  WAIT
  GOTO TEMP
ENDIF
ENDIF
IF 'D' = ANS
REPLACE MAIL:LIST WITH ' ''
DELETE
PACK
USE PMAIL INDEX PMAIL
FIND &MAIL:NAME
STORE # TO TEMP
IF TEMP = 0
  IF FILE(MAIL:NAME)
    DELETE FILE &MAIL:NAME
  ENDIF
  TEXT
  THE MAIL LIST AND DEFINITION HAS BEEN DELETED.
  ENDTEXT
  WAIT
ELSE
STORE T TO UPTOMAIL
ELSE
  TEXT
  SCREEN WAS DELETE, YOU ARE NOW AT THE TOP OF THE DEFINITION.
  ENDTEXT
  WAIT
ENDIF
ENDIF
IFDEF
IF 'F' = ANS
STORE # TO TEMP
SKIP 1
STORE # TO TTEMP
IF TTEMP = TEMP OR. MAIL:NAME = MAIL:LIST
  TEXT
  YOU ARE AT THE END OF THE MAIL LIST DEFINITION.
  ENDTEXT
  WAIT
  GOTO TEMP
ENDIF
THE FOLLOWING SCREENS ARE THE MAIL LIST DEFINITION. THEY ARE JUST LIKE THE PERSON INFORMATION SCREENS.

ENTER IN THE APPROPRIATE FIELDS, VALUES WHICH YOU WOULD LIKE MATCHED. NOTE: ALL FIELDS ARE VALID. THE VALUES ON ONE SCREEN ARE AND'ED TOGETHER. THE INDIVIDUAL SCREENS ARE OR'ED.

ALL DUPLICATES IN THE RESULTING MAILING LIST ARE ELIMINATED
HIT ANY KEY TO CONTINUE

ENDTEXT
WAIT
ENDIF
IF "M" = ANS
STORE "MODIFY" TO ANS
STORE T TO UPTOMAIL
ENDIF
IF "P" = ANS
STORE "PRINT" TO ANS
STORE T TO UPTOMAIL
ENDIF
IF "T" = ANS
STORE "TYPE" TO ANS
STORE T TO UPTOMAIL
ENDIF
IF "Q" = ANS
STORE T TO UPTOMAIN
ENDIF
IF "X" = ANS
QUIT
ENDIF
ENDDO
RETURN
B.17  SEEPEBSO

* SEEPEBSO
* RUNS THE MENU FOR THE PERSON SCREEN
* CALLED BY: PEOPLE
* CALLS : DISPERSION - DISPLAYS THE SCREEN.
* SEECOM - DISPLAYS THE COMMENTS
* FOR THE CURRENT PERSON.
* DISPLAYS COMMAND MENU, AND PROCESSES THE COMMANDS.
*
*
STORE ' ' TO ANS
DO WHILE .NOT. UPTOMAIN .AND. .NOT. UPTOPOPLE
  IF 'L' # ANS .AND. 'R' # ANS
    DO DISPERSION
  ENDF
STORE ' ' TO ANS
ERASE
  @ 3,10 SAY 'PERSON COMMAND MENU'
  @ 4,10 SAY 'TYPE THE LETTER OF YOUR CHOICE' GET ANS PICT '!'
  @ $+2,10 SAY 'C - <DEFAULT> CURRENT PERSON INFORMATION SCREEN'
  @ $+1,10 SAY 'P - ENTER, MODIFY, OR SEE INFO. FOR A PERSON'
  @ $+1,10 SAY 'B - MOVE BACKWARD ONE *** RELATION IS ORDERED'
  @ $+1,10 SAY 'F - MOVE FORWARD ONE *** BY '
  @ $+2 SAY INDEX1
  @ $+1,10 SAY 'L - LIST OTHER EMPLOYEES OF SAME COMPANY'
  IF COMMENTS # ' '
    @ $+1,10 SAY 'R - ENTER REMARKS. PHONE CALLS, LETTERS'
  ENDF
  @ $+3,10 SAY 'Q - QUIT TO MAIN MENU'
  @ $+1,10 SAY 'X - EXIT SYSTEM (TYPE "DBASE MAIN" TO RETURN)'
READ
  IF ANS = 'B'
    SKIP -1
  IF # = 0
    GOTO BOTTOM
  ENDF
ENDIF
IF ANS = 'F'
  SKIP
  IF EOF
    GOTO TOP
  ENDF
ENDIF
IF ANS = 'L'
  ERASE
  STORE # TO LISTNO
  STORE ORG:NAME TO LISTTABG
IF INDEX1 = 'LAST NAME'
USE PEOPLE INDEX PCOMPANY
ENDIF
TEXT
LAST NAME  FIRST NAME  MIDDLE  NICK  CITY
-------------------  -------------------  -------  -------  -------
ENDTEXT
FIND @LISSTARG
DO WHILE ORG:NAME = LISTTARG .AND. .NOT. EOF
DISPLAY OFF FIRST:NAME,LAST:NAME,MIDD:NAME,NICK:NAME,CITY,STATE
SKIP
ENDDO
IF INDEX1 = 'LAST NAME'
USE PEOPLE INDEX PNUMBERS,PNUMCOMPANY,PNUMBER
ENDIF
GOTO LISTNO
WAIT
RELEASE ALL LIKE LIST*
ENDIF
IF ANS = 'P'
STORE T TO UPTOPEOPLE
ENDIF
IF ANS = 'R' .AND. COMMENTS # ' '
DO SEECOM
ENDIF
IF ANS = 'X'
QUIT
ENDIF
IF ANS = 'Q'
STORE T TO UPTOMAIN
STORE ' ' TO ANS
ENDIF
ENDDO
RETURN
B.18 \textbf{UNPACK}

* UNPACK
* DECODES THE RELATION TUPLE INTO THE DISPLAYED VARS.
* CALLED BY: DISPERSON
* CALLS : NONE.

* IF GEOG:AREA = 'O'
  \quad \text{STORE 'X' TO PB33}
ELSE
  \quad \text{STORE ' ' TO PB33}
ENDIF

IF GEOG:AREA = 'Y'
  \quad \text{STORE 'X' TO PB32}
ELSE
  \quad \text{STORE ' ' TO PB32}
ENDIF

IF GEOG:AREA = 'N'
  \quad \text{STORE 'X' TO PB13}
ELSE
  \quad \text{STORE ' ' TO PB13}
ENDIF

IF GEOG:AREA = 'L'
  \quad \text{STORE 'X' TO PB12}
ELSE
  \quad \text{STORE ' ' TO PB12}
ENDIF

IF GEOG:AREA = 'T'
  \quad \text{STORE 'X' TO PB11}
ELSE
  \quad \text{STORE ' ' TO PB11}
ENDIF

IF CONT:TYPE = 'S'
  \quad \text{STORE 'X' TO PB20}
ELSE
  \quad \text{STORE ' ' TO PB20}
ENDIF

IF CONT:TYPE = 'P'
  \quad \text{STORE 'X' TO PB19}
ELSE
  \quad \text{STORE ' ' TO PB19}
ENDIF

IF CONT:TYPE = 'C'
  \quad \text{STORE 'X' TO PB21}
ELSE
  \quad \text{STORE ' ' TO PB21}
ENDIF

IF MEMB:TYPE = 'I'
  \quad \text{STORE 'X' TO PB24}
ELSE

STORE ' ' TO PB24
ENDIF
IF MEMB:TYPE = 'R'
STORE 'X' TO PB14
ELSE
STORE ' ' TO PB14
ENDIF
IF MEMB:TYPE = 'C'
STORE 'X' TO PB15
ELSE
STORE ' ' TO PB15
ENDIF
IF MEMB:TYPE = 'N'
STORE ' ' TO PB16
ELSE
STORE ' ' TO PB16
ENDIF
RETURN