DESIGN OF A PERSONAL GENEALOGICAL DATA BASE SYSTEM

by

MARY JO BIRD

B. A., Fort Hays State University, 1976

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1981

Approved by:

Major Professor

TABLE OF CONTENTS

		Page
I.	INTRODUCTION	1
	1. Motivation	1
	2. Problem Description	2
	3. Definitions	6
	4. Data Base Terminology	9
II.	DEVELOPMENT OF THE DATA BASE	11
	1. Overview	11
	2. Research Log	12
	2.1 Document Number to Search Number	12
	2.2 Document Number to Date	15
	2.3 Search Number to Date	16
	2.4 Dependencies	16
	3. Pedigree Charts	17
	3.1 Unique Numbering	17
	3.2 Record Form, Binary Relations and Associations	19
	3.3 Relations Based on Common Retrievals	21
	4. Family Group Worksheets	26
	4.1 Using the Family Group Worksheet in the Data Base	26
	4.2 Breaking the Record Form into Relations	31
	4.2.1 Separating the Dependencies	31
	4.2.2 Placing Domains in Other Relations, Including a New Relation	32
	4.2.3 Retaining Information Through Creation of a New Relation	33
	4.2.4 Refining Other Effected Relations	34
	4.3 Handling Non-family Persons	35
	4.4 Maintaining Children Information	36
	5. Document File	38
III.	DESIGN OF THE DATA BASE	41
	1. Justification of the Model	41
	2. Normalization	43
	3. Relations and Dependencies	49
	4. User Interface	51
	5. Implementation Considerations	57
IV.	CONCLUSIONS	60

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

THIS IS AS RECEIVED FROM CUSTOMER.

Table of Contents (Continued) Page

1. Results 60
2. Future Research 61

APPENDIX

1. Research Log A-1
2. Pedigree Charts A-2
3. Family Group Worksheets A-3
4. Cross-Referencing the Files A-4

LIST OF FIGURES

Page

2.1 Resulting Research Log Dependencies 18

2.2 Example Transaction 23

2.3 Relations for Producing Pedigree Charts 25

2.4 Resulting Pedigree Chart Dependencies 27

2.5 Resulting Family Group Relations 39

3.1 Relations in Third Normal Form 47

3.2 The User Interface 58

ACKNOWLEDGEMENTS

I would like to thank Dr. Elizabeth Unger for her guidance, suggestions, and tireless editing. I would like to thank Drs. Paul Fisher and Rod Bates for serving as committee members. Thanks also to Karen Dungey, for her editing and dedicated typing of the manuscript. I am indebted to Mrs. Golda Sitz and Mrs. Agnes Elzinga, for helping me to understand genealogy and its possibilities for computerization. Finally, thanks to my husband, Frederick, for his support and encouragement.

INTRODUCTION

CHAPTER I

1. Motivation

In-home computer systems must be simpler than current commercial systems, responsive to the user, direct in their application and accepting of varying types of input. The non-professional general-population aspect of the new computer users must be taken into account. This has been done in game cartridge programs and a few other preprogrammed packages. But for the user, there is still a feeling of lack of control in making the programs do "what we want them to." There is also little provision for variable information. Storage for anything at all, let alone a whole "file system," must usually be obtained separately at an additional price. And then, the few who can use these capabilities are those that already know, or are willing to learn, how to program. My contention is that the need exists for a home data base management system.

A few years ago, there was the expectation of a boom in home computers. What followed was instead a boom in small business microprocessors and single-purpose process controllers. Still, there has been innovation in the home computer market. But "for those wanting to retrieve information. . .from a data bank, a (necessary) telephone modem will cost around $200."[(20)] There is a long way to go before computers are commonplace home appliances.

One big problem is user acceptance. There remains a computer-human understanding problem. This is a two-way misunderstanding. The interactive "dialog" produced by a computer is usually rigid, unvarying and completely predictable. It can border on the irritating, especially when it doesn't seem to understand what the user wants. The computer "expects" the same patterns from humans. It becomes meaningless to communicate with it about any other matter than it is currently concerned with, or in any other manner than in the precise

unforgiving query language it uses. Any person might wonder why they cannot understand a machine developed by other people. One approach is rigorous training in the public school system on "how to interact with a computer." But the solution to the problem of user acceptance should lie with computer professionals. Designers and programmers need to take another look at the public interaction required of their products.[13][18]

On-line home systems must be simple to understand and use. These systems must be lenient in their acceptance of input, allowing free-form presentation of at least regular-grammar commands. And, if all else fails, the computer must react in some other manner than merely responding that an error has occurred.

My purpose is to show the in-home usability of a computerized "file system." There is no need to create an application where none exists. That tends to place the computer in the realm of luxuries. Instead, my approach is to enter the world of a group of people who already have working file systems in their homes. Each of these people uses his or her file independently. Some have used the computer during the creation or maintenance of their files. The people involved are genealogists. Sometimes they start as hobbyists with little plan for, or need of, a file. But for many, genealogy is a way of life. A good filing system to organize and control their records is a necessity. There are presently many ways of building, using and maintaining these home files. The basis for a computerized system could be taken from several different filing methods.

2. Problem Description

The problem is to design a data base. The data base will contain genealogical information for one user (or family of users) at a time, organized in a meaningful way. The data base will be kept on a home computer for use by a genealogist. The data base will help to organize the records and data obtained

from the user's research and personal records. The logical organization will be derived from a currently used non-computerized genealogical file system, so the organization will be familiar to a genealogist. By "meaningful" above, I mean the organization will be in terms of the filing and retrieving operations the user currently uses or wishes to use, in a set of files.

Problems confronting the genealogist are these: organizing a large collection of personal and research records; organizing ancestor data so it is easy to go forward and backward (chronologically) through them according to family lineage; getting the most from any research done; saving time recording information, and avoiding re-recording or producing redundant information since it is not easy to update; avoiding recopying all information from one's files to computer cards; cross-referencing files appropriately and determining a storage order to use to facilitate quick retrieval, and using an organization that will be meaningful to one's descendants so that they can use the information obtained and/or carry on.

Computerization problems include the following. Record forms currently used may need to be formatted and stored by the computer. Organizing family information seems at first like a tree or even a binary tree problem. The "family tree" was where a lot of the terminology and ideas for trees in programming came from. Yet, upon closer examination, the answer is not so simple. Although it's not a binary tree from parents to children, it can be from child to parents. This representation is often used in a genealogist's pedigree charts. But there are not always only two parents for a child. A child may be adopted or a parent may remarry. Information on step-parents should be kept too, with appropriate linkages. Actually, it wouldn't be a binary tree even if only two parents were considered. A binary tree has only one inward branch for any node. To account for siblings, each child of a parent should have a branch inward to that parent.

Perhaps there should be sibling-to-sibling branches. There are several branches from a node: sibling, spouse, children and parents. There is always the possibility that there are two or more of each of these. It can get very complex. Besides the problem of multiple values, there's the possibility that some of this information is not yet known. Does one create blank "slots" for persons or other information one expects to have later? Then a distinction must be made between information believed obtainable and information "possibly" obtainable. Another problem is to maintain all the linkages. If there are parent-to-child links but not vice versa, it will be possible to trace lineages forward but not backward. If there is a child-to-father link but not one to the mother, it may still be impossible to trace backward unless there is always a husband-to-wife link. Then how is it verified that the wife in the link is the one who is this child's mother? Also, should the child be linked to all step-siblings in addition to full siblings? Another consideration is the unique-identification problem presented by those people who were named after a relative. This is fairly common, especially the more generations that are included in the research. There must be a way to distinguish among several people with the same name if their birthdays are not known.

If the underlying data structure is not a tree, there is still no easy way to model the data to solve the above problems. The possibility of a set, "mother, father, child, child. . ." doesn't seem like a good relation or an easily manipulated hierarchy. Even if there's only one child per relational "tuple," it wouldn't be easy to retrieve all children for a family. A solution is to base record types on current forms of some filing system. There are many forms and systems available.

Another important consideration about this data base is that it is a working tool. It will be changing and growing as long as it is used. Most of the

changes will be additions. As a repository of ancestral information, it will grow as more research is done, also as marriages occur and more children are born. An "entity" is not removed due to divorce or death. Very few deletions will be necessary. Additions will be commonplace. Modification of existing records will be necessary when new research is done for an individual or family previously researched, or when data is found to be incorrect. Possibly occassional reorganization is also necessary.

Some computerization has already taken place in genealogy. The most widely known is at the Latter Day Saints' Genealogical Library based in Salt Lake City, Utah. Records were placed on microfiche to decrease the amount of storage needed and computerized to provide quick access to all the information available. Known as CFI (Computer File Index), the system provides indexing to 50 million names.(9) At least one independent company offers to search these files for a customer. A second system is the "Computerized 'Roots' Cellar" of The Everton Publishers, Inc. It maintains records on ancestor information submitted, and names and addresses of the submitters. This information is all made available to others who might be interested in the same ancestor and wish to correspond with a submitter. A third system is offered by Accelerated Indexing Systems, Inc., also of Salt Lake City. They offer six different computer searches through sets of names input from government and other historic records.(10) Each of these systems can be called a data base. But they are all concerned with helping a researcher find information; their purpose is not to organize and manage the researcher's personal files. The purpose of the data base designed in this report is to organize and manage an individual's genealogical records.

3. Definitions

The following definitions and clarifications provide information on the usage of terminology in this report.

Document: a record of an event, transaction, person or family. Usually it is in the possession of the researcher. It includes deeds, certificates, receipts and many other things that contain family information.

Family Group: consists of parents and children for one marriage only. A remarriage of a parent is considered a separate family group. A genealogist records the information known about each family group on an individual Family Group Worksheet. Some family group forms also have room for information on the grandparents. In this report, grandparent information is assumed to be a part of another family group. There are two uses for family group sheets, a working record and a final record. During research, notes from various sources are put on family group worksheets, keeping information from different sources on different sheets. A large number of worksheets per family group may be obtained. Eventually, a final family group record will be produced, incorporating all this data. Where contradictions have been found, data from different sources is compared and evaluated. The information that is most likely true is placed on the permanent record. When the family group worksheet (or the final record) is referred to in this report, it will be abbreviated "fgw." (See the Appendix for further explanations.)

Files: separate categories the genealogist uses to organize the documents and forms in his possession; or the separate folders, file drawers or file cabinets where these papers are kept. Possibly each type of form used for research data will be kept in a separate file. There are many record forms and filing systems in the literature. This report will concern itself with one system and its forms. Each form is a file category - or file. The files in the

selected system are Family Group Worksheets, Documents, Pedigree Charts and Research Log. The system includes cross-referencing among the files.

Lineage: family ancestry and descent. It may include only oneself and one's parents, grandparents, great-grandparents, etc. This is a backward view of lineage, and is concerned with direct ancestry. Or, it may be a forward view of the family starting with a common ancestor and including all descendants. Tracing lineage may include one or both approaches.

Log, Research Log or Research Calendar: a summary form to record research steps, as they are taken. Date, condition of the source and location are some of the data recorded. A description and complete identification of the volume researched is recorded so the same information doesn't have to be rewritten on all the notes taken from this source. This saves a lot of time and effort when each note is placed on a separate card, sheet of paper or even family group worksheet. The complete identification of the volume makes it possible to easily locate the source at a later date, or by another researcher. The log also serves as a bibliography of research done. Further information is available in the Appendix.

Marriage Identifier or Marriage ID: a unique identifier for family group. This is not used by genealogists, but there is a need to easily identify families. All the family group sheets on a specific group are filed together. But it is not easy to distinguish sheets on two families where the fathers of both had the same name, without reading the worksheet. Also separate marriages for the same person need to be identified. Marriage ID has several uses in the data base developed.

Names: a major consideration for a genealogist. The management of the names that data is known for is second only to the management of records as an organizational problem. One method is to group records in a file by surname,

then alphabetically by given name, then by birthdate.[15][16][17] (There may still be several records with the same surname, given name and birthdate which the user may or may not order further.) The researcher must control not only a multitude of individual names, but also a large set of surnames. This is not obvious as the beginner is interested in one or two family names. In tracing the lineage backward, it soon becomes necessary to accommodate a large number of additional surnames for married women. For each generation traced, the number of surnames doubles. This means for any additional preceding generation, one must add as many surnames as are already represented in the files.

Numbering: gives a method of identifying and cross-referencing individuals. There are several methods used. For instance, numbering may be sequential by birthdate or by time of addition to the files. The unique number concept and domain is used often in the report. Numbering is further explained in the "Pedigree Charts" section of Chapter II.

Pedigree Chart: a graphic display of family relationships. The forms available span varying numbers of generations and allow varying amounts of information on each person. Photo charts are even available. There are two basic forms of charts, and two sets of charts kept in a system. The two forms correspond to the two approaches to tracing lineage. The two sets are working and final pedigree charts. As research is done, pedigree information is entered onto working charts. All data is placed there, even if it is suspect. The final charts are reserved for only data that has been verified. Creation of both sets is an on-going process. Any histories or other reports concern only the final set, although anything from the working charts that was later dismissed may be included with reasons why it was rejected. The Appendix contains further information.

Primary or Original: first-hand accounts or records produced at or near

the time of the event recorded. These are very valuable to the genealogist. They are the main records sought in research. In the subsequent evaluation and analysis, primary records are the best and most conclusive evidence to base judgments on. Their uses include a basis for judging contradictory data, and supporting evidence for claiming a certain person as an ancestor.

Record forms and Records: There are many. Family group worksheets, pedigree charts and logs are widely used for storing and organizing data. They are used in the design. These and personal documents constitute a large part of a genealogist's records.

Record Category or Record Group: book or a collection of recorded information with a collective title. Just what constitutes a record category can vary with the subject of the collection. "'Record group,' (is) a term used in archiva institutions."(17) A record group is determined by all the descriptive levels needed to find the "record" again.

Search: research when limited to a single record category. The effort, process and results of looking through a set of records or a volume for family information, are also called searches. How much looking is done in a search varies as much as the amount of information obtained does.

Secondary: second-hand accounts or records, including copies of primary records (usually). They may be less accurate than primary, but out of necessity are often the basis for information in a researcher's files.

Source: the object of a search. It is used in the usual literary sense of "the place where information was obtained."

4. Data Base Terminology

"Domain" is used in the relational sense, as a component in a relation. When two or more components can have values from the same set, domain is used to

entail the distinct meaning a component has within the relation.[8] "Domains" refer to parts of a record type in the data base. The corresponding term in the record forms used as a basis for the record types is "field." It means a named place, within the record form, for putting data. The term "record" is used in two different ways. In the definition of record above, the genealogical meaning is given. Also used is the data base meaning of "an instance of a record type." Usually the meanings will be distinguished by use of "record forms" and "record types," respectively. "Relations" is another word used in two ways. Often "relations" and "record types" are used interchangeably to describe an entity set in the design. The record types developed in Chapter II could, however, easily be thought of as existing in a network. So the record types are not really limited to relations. The other meaning for "relation" is just the usual usage - being in the same family. Another word with two meanings is "search." The definition above describes the genealogical term, and the entry in the research log that is assigned a search number. This same search number domain is used in other record forms and record types. "Searching" is also used to describe the computer processes of locating certain information within the data base.

5. Organization of Report

The first chapter contains motivation for the project, a description of the problem and definition of terms. The second chapter contains details of the problem and describes the process used to develop the data base design. The third chapter gives the design and describes the user interface. The last chapter summarizes the results and suggests future work.

CHAPTER II

Development of the Data Base

1. Overview

The complexities of family relationships and variations would carry over to a logical data base design based on them. A simpler and more appropriate approach is to base the design on a file system already used by genealogists. There are many such systems. One system is to keep a bibliography, a "master list of references, numbering each source,"[16] and notes obtained from research Each note has the number of one of the sources. This is simple and easy. But many record forms routinely used by genealogists are not included. Another system[14] uses a research calendar or log, fgw's, pedigree charts and a document file, all cross-referenced to each other. The type of forms used are commo in the field, and this system is also simple. A third system incorporates the files into a comprehensive "'central file' in the home where all family papers are housed."[17] All records are kept in the same physical file, and classified by surname, major locality, minor locality, then record category. It is not a specifically genealogical system, and does not limit the records to any standard formats. In effect, the number of domains is infinite and the dependencies undeterminable. These three systems stand out in the literature as being simple straight forward and understandable even to someone unfamiliar with the files so organized. The second filing method was selected.

The system is documented in "Family History for Fun and Profit," formerly known as "The Jurisdictional Approach."[14] There are three files using standard forms and a file for documents that doesn't immediately seem computer representable. This chapter contains the explanations of form fields and associations necessary to develop relations from each of the four files. The

real-world meaning of binary relations represented by data on the forms is also examined. The relations are normalized in Chapter III.

2. Research Log

An example Research Log is shown in the Appendix, along with the relational dependencies. There are many multi-valued dependencies. Some of these are removed by examining the meanings of the domains represented by the record fields in the following discussion.

2.1 Document Number to Search Number

A search is a single research event. It is the examination of a set of related historic records. The relation document number : search number actually means document number : source. Any search will cover a source or set of sources. There are three different meanings that could be given to the document : search relation. (1) For every search, there is a possibly null set of documents containing pertinent information. (2) For every search, there is a set (again possibly null) of referenced documents. (3) For any document, there is a unique search number that corresponds to it, but a search does not always involve documents. These meanings are represented by the following: (1) search number →→ document; (2) search number →→ document number; and (3) document number → search number, and indeterminate in the reverse direction. What a search is, is somewhat open to interpretation. It could be that every source - book, etc. - inspected is given a search number; every source is a search. Or perhaps every set of related records, every group of similar types of information - possibly spanning volumes - is a unique search. "Each record category consulted is a separate search. Number each search chronologically by date of search. . ."[14] The relation between document and search number becomes a question of the relation between document and source. A document in the document file is a source

possessed by the researcher. A document is a held source. Any other source is a non-held source. I am making a distinction which is only implied in the literature referenced, but which will become important later.

Taking the first meaning above for document : search, the results are unfavorable. When searching through a book, every time information is found that may verify, augment or even contradict information on a document in the researche possession, an entry is created on the research log with the number of the document, the search number and text as to what information it is that is referred to in both the source and document cited. This will be called "loose cross-reference." This is probably not what is meant by the authors in setting up the document number, search number, text entry. Also, it would be unnecessarily tedious and unclear (as to what constitutes a loose cross-reference). The extra amount of writing required (rewriting a search-source entry for every document so there will be a key for each entry) would not be offset by much additional information. It is most probable that many sources encountered and deemed relevant to one's cause will have some connection to one, some or many documents held, if "connection" means they both happen to pertain to the same family, person or event in any manner. It would not be especially helpful to cross-reference this source - or search number - to all such documents. If one had to write down all such information on the log, it could be time-wasting and voluminous. It would defeat the purpose - i.e., as a summary form - of the research log. Also it would imply that one always had available a list of the documents held, or could remember what each entailed, any time research is conducted. This is an unnecessary burden. Therefore, search number → → (loosely cross-referenced document number) is dismissed as a relevant or practical relation.

There is the possibility that when entering information on the log, instead of complete separate entries for each document cross-referenced during a search,

the repeated information could just be "dittoed." Then the generation of complete separate entries could be left to the computer. However, this merely postpones the problem, because in a relational data base, there would be a lot of redundant information.

Meaning (2) above is, for every search there is a set of referenced documents. It's the opposite extreme from meaning (1). When cross-referencing a search number with a document, let that mean that the document is specifically named or referred to in the source. Then the relation search number → → document number would return valuable information if it returned anything at all. For any search done, one would get back the identification of any documents that had direct bearing on the information obtained. However, it is highly unlikely that a source information is obtained from will specifically name or refer to a privately held document. This is the case unless the researcher uses a copying machine frequently and keeps the "xeroxed" copies of original source records in the document file too. Then the document number(s) for a search may actually be of reproduced copies of pages in the source that was searched. This is a possibility. But it's also possible to restrict the document file to only original documents. This will be done here for simplicity. Since search number → → (directly referred to document number) is not probable, using it in the data base would not reflect a useful real-world relation.

The third meaning is, for any document there is a unique search number but not necessarily vice versa. To examine this, I return to how the ideas of document and source are related. The use of a document in a system is that it contains pertinent information. The use of a source researched is the same. The two are similar except that one is possessed and one is not. From now on, "document" will mean a held source and "source" will mean a non-held source. Any source used always has a search number associated with it. A document will

too when the data from it is placed on fgw's or pedigree charts. Usually, searches will involve sources. But occasionally a search will be on a document. The two can be kept separate. If the search number is associated with a document, there will be a document number in the log entry. If it is associated with a source, there will be no document number, but there will be an identification of the source in the textual description. And, since search number uniquely identifies sources, the user would want to retrieve search number instead of document number in a query based on search identification. But this would be a meaningless operation: search number would be both the key of the query and the object returned. So if a document was searched, then search number → document number. But for a source, search number → search number. Conversely, always document number → search number.

From all three of the possible meanings examined, the conclusion is that search number → document number is a pointless operation to pursue. But document number → search number is useful.

2.2 Document Number to Date

Given the previous arguments on document number and search number, the only time a particular document number is present on the research log is when that document is the object of the related search. Presumably this would entail only one search date. Usually when one takes data from a held document, it doesn't take more than one day: document number → date.

This could be changed to "always" if one restriction is observed. It is possible that it may take more than one day to finish a search involving one document. If so, the dates would still be few and close together. When this does happen, one could record the search on the log by the date the search began, rather than under a string of consecutive dates. This saves redundant recording and always gives document number → date.

2.3 Search Number to Date

The above restriction keeps the number of recorded dates for a search on a document to one. Possibly the same thing can be done for a source. It is more likely that a search on a source will take more than one day. There are some sources that are so voluminous, one can not possibly hope to obtain all the information in one day. The only reason it is likely that a source is more voluminous than a document is that the average person doing family research has single-record documents and doesn't own a lot of personal family history volumes, but does have access to such volumes in other places. Still, for any search that is conducted, the source contains a finite set of information, gatherable over a finite, well-defined set of dates. If the interpretation of "source" is restricted enough, a search of that source will only last for a very small set of dates. At the most, search number $\rightarrow\rightarrow$ date. If the dates are few enough, and close enough together, information is not lost by recording the search with only the beginning date of the search. Then search number $\rightarrow$ date.

Whether it can always be assumed that the dates for any given search are few and close together, goes back to the interpretation of "search." As long as the object of a search is restricted to one volume or set of records of an arbitrary fixed size, then one can use the assumption. Indeed, if a volume or set is larger, the researcher can assign multiple search numbers. This also keeps the amount of information obtained from any search manageable.

2.4 Dependencies

Other research log dependencies are:

date $\rightarrow\rightarrow$ search number
date $\rightarrow\rightarrow$ document number
search number $\rightarrow$ document number
or search number

The key would appear to be document number, if it were always present, which it

isn't. In normalizing, it will be useful to separate the research log into two record types, one for documents, the other for sources. This will be discussed later. Document number is a key for the document log, and search number is a key for the source log. Dependencies are shown in Figure 2.1.

3. Pedigree Charts

3.1 Unique Numbering

Unique numbers are used in various ways by genealogists, but none comprehensively. Usually each individual on a direct ancestors' chart in the pedigree file is given a number. The foremost person in the lineage is number 1, and his father and mother are numbered 2 and 3. The father's parents are 4 and 5; the mother's, 6 and 7, etc. The father of anyone on the chart has a number that is twice the child's number. A person's mother's number equals two times the person's number, plus one. Each chart in the file can also have an identifying number; then referencing a person consists of a combination of chart number and location on the chart.[24] One person can have more than one number, if appearing on more than one chart. An example of this is a person in the last generation shown on a chart is often the first person on another chart. Unfortunately, this numbering system is often used only on the pedigree charts. On other record forms, the person is referred to by name but not always by number.

Another system relates a person to direct ancestors. The oldest ancestor in the files is number 1. Each of his or her children are numbered 1.1, 1.2, etc. The children of 1.1 are 1.1.1, 1.1.2, etc. A person is thus connected to a parent, a grandparent and so forth. This system is used after the research is complete enough to write a family history, not during the research. It can be used in addition to the numbering on the pedigree charts, but there is no fixed correspondence between the two. Another system uses a combination of generation and individual numbers:

Figure 2.1 Resulting Research Log Dependencies

Document number → search number, date, description, locality, time period, result, surname[1]

Search number → date, description, locality, time period, result, surname[1]

Search number : document number is indeterminate[2]

Date →→ search number, document number, surname, description, result[3]

Date : locality, time period is indeterminate[4]

1. The limitation of surname to one value per entry is discussed later in this chapter.

2. Some searches do not involve documents.

3. Description and result can have multiple values because every search number has a unique description and a result associated with it.

4. Any date may cover several searches, all such searches are usually in the same locality and may be for the same time period historically. So, it's possible these are one-to-one.

> "A numbering system, completely different than that used in tracing your line back on working (pedigree) charts, is used to give clarity to your written genealogy. Each person in the family line is identified by an individual number, assigned consecutively, and by generation number."(3)

The generation number is appended as a superscript. The main problem with using either of the last two numbering systems during the research is the updating necessary when a new generation of "oldest" ancestors is added. Addition of a new number 1 ancestor would require extending a string of digits and periods on the left for every person already in the data base. In the second method, generation numbers for all people would have to be increased by one.

The easiest method of numbering would be to assign a unique number to a person when information on him is first entered into the data base. Since a person entered is rarely removed, there is no need to reassign or deallocate numbers. Thus, these numbers are not physical storage keys. The numbers would be known to and used by the user without data dependence. The numbers would also help the user to manage and solve the confusion of all the names data is maintained for. The numbering would be random with respect to generations and lineage. Therefore, adding a new generation at either end of the spectrum would not require any number updating.

3.2 Record Form, Binary Relations and Associations

The relations developed from pedigree charts apply equally well to working or final charts. However, the final charts, as a permanent record, should be kept separate from the data base. They are not changing, time-dependent records, as the working charts are. Therefore, it is assumed that the charts referred to here are working pedigree charts.

There is a lot of information that can be kept on individuals. One reference lists 25 categories of "basic information" with numerous sub-categories, including even such items as political affiliations, income and wealth during

lifetime, forms of recreation and achievements.[16] Most of these are not considered here because they are associations of the person entity set similar to the ones which are included. The data could be kept mostly in the family group file, or in the pedigree file, or spread evenly in both. Some view the family group file as the repository of most note taking[14] and therefore of most data on individuals. Some view the pedigree chart as a skeleton form to show relationships rather than describe those included. It is "a visual aide, not a complete record of those included in it."[15] Maybe most information should be kept in the family group file. For now, the question will be abandoned. In the following discussions of the relationships in each, and in the normalization process, the issue will partly be resolved and partly become irrelevant.

The information a pedigree chart contains and dependencies it represents are shown in the Appendix. A set of fields for the pedigree chart is taken from reference 14. For each person, a name, birthdate, birthplace, death date, place of death, marriage date and occupation can be maintained. Also, there's additional information implied in the lines of standard charts: spouse, child, mother or father and sibling connections. One other field, a unique number, is included to provide the retrieval capability explained earlier in this chapter.

The dependencies here are simple because data values like birthdate, marriage date or occupation can not determine any of the others. Name doesn't even determine any of the others because within a family there may be two or more people with that same name. At first, it seems that for all married people, a spouse would uniquely determine the name or other identification of a person. But this is not true since anyone may have married more than once. In fact, only the unique number determines any of the other fields, and it determines them all. So, unique number is the only key.

It is possible to keep all the value fields together in one relation, but it

is not obvious how to store all the connection information. If all in the same relation, fields are needed for spouse, child, mother and/or father and sibling. There are several reasons why these shouldn't all be in the same relation. Some people have no spouse, or no child, or no siblings. The fields would have to be given null values. Some people have had more than one spouse, child or sibling. This could be solved with multiple records per person, but this would cause a lot of redundant information in the file. Because there are three domains with multiple values, the update, deletion (when deletion is necessary) and addition anomalies would be even worse than if there were only one. The problem of how to construct a chart for a person with multiple spouses is not addressed by such a single relation.

3.3 Relations Based on Common Retrievals

The record form must be broken up. How to do it and which connections to place in record types, have many possible answers. The following discussion considers what retrievals are to be performed on this information, in order to design relations that are useful and appropriate.

Genealogists use pedigree charts to show lineage and family relationships. There are two methods for charting.[14] The direct ancestors chart goes backward with respect to time. The most recent person is first, with connections to direct ancestors only. This chart doesn't show siblings. It is a binary tree on the standard forms.

The common ancestor, or descendants, chart goes forward in time. The common ancestor is first. Lines from him/her lead to his/her children, and lines from each of them lead to their children, etc. This chart doesn't always show spouses.

The end design should have the capability to retrieve information corresponding to both methods. Genealogists have said it is important to be able to

work in either direction.[3][14] In many filing systems, there is no easy way to do one or both.

On the following page is an imaginary interaction to retrieve a descendants' chart. Hopefully, when the parents and children are retrieved, their names were placed in appropriate places in a formatted set of records, for later printing when the transaction is complete. One thing that emerges about this type of transaction is the recursive nature of the processing. The same type of information and connections are retrieved and followed first for the person named in the query, then for each of his children in turn, and meanwhile, for each of their children.

To do this kind of processing recursively, the following order of searches and retrievals is helpful. Use the name in the query to get to a unique number. If more than one unique number is returned, ask the user to make a choice. Use the unique individual number to get to all spouses. If more than one, ask the user to make a choice. Use the first unique number and the unique number of the chosen spouse to pick the proper marriage (family group). Use the identification of the marriage to retrieve all children of that marriage. For each child retrieved, go through the same procedure, from selection of spouses. Hold every name (and possibly other information) in a formatted file for later printing. When the proper number of generations have been retrieved and spanned, conclude the transaction.

The information to be retrieved is name → → unique number; unique number → → numbers for spouses; (male unique number, female unique number) → marriage ID; marriage ID → → children. Those after the first are recursive. Figure 2.3.a contains relations that allow the above retrievals.

A modification will make it easier to search for all spouses, an operation that will be done frequently. As it is now, when the search is made, just

Figure 2.2 Example Transaction

INPUT	RESPONSE
Descendants of Thomas B. Morris for 3 generations	
	Thomas B. Morris Married
	Maggie Jo Dallas
	***
	There are 8 children
	***
	Leonard Morris Married
	Gladys Morris
	***
	There are 2 children
	Children obtained.
	***
	Gladys Morris Also married
	Hugh Church
	Shall the above be included?
Yes	
	There are 1 children
	Children obtained.
	***
	Next child of Thomas B. Morris
	Merle D. Morris Married
	Gladys Morris
	***
	There are 1 children
	Children obtained.
	***
	Next child of Thomas B. Morris
	Paul Morris Married
	None
	No children
	***
	Next child of Thomas B. Morris
	Stanley C. Morris Married
	Mary P. Morris
	Blanche R. Morris
	Which marriage shall be followed?
Both	***
	Children obtained.
	.
	.
	.
	End of 3 generations.

finding the number of the known person in the marriage relation requires searching both unique number domains. Addition of a male/female flag to person makes it known for any person, which domain to search. This eliminates one useless search and would save, on the average, half the time for this retrieval. See Figure 2.3.b. The domain for the identifier of child is actually the same as the unique numbering system used in the other two relations. It is a key in both family and person. Since it is the only key in both, the two relations can be combined. See Figure 2.3.c.

Comparing these relations with the pedigree chart file (see Appendix) that was the basis for this development, shows they are compatible. These have the extra marriage ID domain. This is similar to the unique number domain, providing unique identifiers for an "entity" set, the family group. Here, it is not really necessary since it could be replaced in person with the numbers of both parents. However, it will be useful later in the family group file. All the extra fields in the original record form - those not included above - except for marriage date, can be incorporated into person. Marriage date can be incorporated into marriage. The original record for pedigree chart has been split into two relations, each one having a single key. (Male number and female number may also be a key for marriage, if two people who married each other twice are recorded as one marriage. This decision depends on how common of an occurrence this possibility is.) Fortunately, the marriage relation makes it possible to account for multiple marriages, and to trace a separate chart for each of a person's marriages.

The type of transaction the previous design was based on, was a forward-retrieving descendants chart. It must be ensured that backward retrieval is also represented by the relations chosen. The direct ancestors method of retrieval follows: Use the query name to find a set of unique numbers in person. If more than one, ask the user to make a choice. Use the unique number of the person

Figure 2.3 Relations for Producing Pedigree Charts.

a.

Person (name, unique number)
Marriage (husband's unique number, wife's unique number, marriage ID)
Family (marriage ID, child)

b.

Person (name, number, m/f flag)

c.

Person (name, number, m/f flag, marriage ID of the marriage he was born into)
Marriage (male number, female number, marriage ID)

selected to obtain marriage ID of his parents' marriage. Use the marriage ID to get numbers of mother and father from marriage. Use each of these numbers to retrieve information on the parents from person, separately. Hold this information in a temporary file for later printing. Also, while retrieving data on the parent in person, retrieve the marriage ID in that record. It is of the marriage to be used in the next level of ancestors. Repeat the previous procedure for that marriage. Keep processing until the requested number of generations have been traced.

Reading through the above procedure while following the actions in the two chosen relations (Figure 2.3.c), shows that the procedure can be executed elegantly. All retrievals except the very first one, are on primary keys. Figure 2.4 gives the dependencies of the relations.

4. Family Group Worksheets

A family group is defined as a husband, wife and their children. If a man has had more than one wife, or a woman more than one husband, each husband-wife combination constitutes a separate family group. It is recommended that one keep information on all of an ancestor's marriages, even though lineages may be traced through only one of them.(14)

4.1 Using the Family Group Worksheet in the Data Base

Queries concerning the fgw include the following: Retrieve all information on a specific family. What documents contain data for a specific family? What research steps (searches) have been done involving a family? For a particular document or numbered search, list the family groups included, or retrieve all family group records effected by information obtained from a particular search.

Family group dependencies are shown in the Appendix. There are several problems to be resolved. First, there are different ways fgw's are used. A

Figure 2.4 Resulting Pedigree Chart Dependencies

Unique number → name, birthdate, birth place, death date, death place, marriage ID for parents

Unique number →→ marriage ID, marriage date, spouse, occupation[1]

Marriage ID → marriage date, husband's number, wife's number

1. Occupation can be limited to one per person. Some genealogists are interested in only the main occupation. This would not be a multi-valued dependency unless the researcher had a need for recording all occupations.

researcher can place research notes directly onto fgw's, using one per source for a single family group. It saves rewriting the notes, but there may be many sheets for each family. What information came from which source is maintained. Determining the source for a single item of information doesn't require repeating the research.[14] All information on a family group is not put together on the same fgw until the research is completed, analyzed and evaluated.

Some authors suggest taking notes on file cards or plain paper, rewriting them later on fgw's. At that time, the notes may either be each written on a separate fgw or collected onto the same sheet when various notes apply to the same family group. The goal is still to eventually place all the information on any family together.

If the design keeps family group information separate by source, then finding all information on a family will be slower. But the integrity of each source is maintained. The key would consist of both a family identifier and a search number, since every source for a family has its own fgw and every source has a search number. On the other hand, if the data base puts all family group information together as it is collected, this will not all have to be done later; and the benefit of seeing all the information together in one form doesn't have to wait until the research is done. Retrieval is quicker since only one record has to be located. As shown later, this retrieval is on a primary key. But the relation of each item of information on a family, to its source, won't be maintained in the data base. This would not create a problem, for two reasons. The original notes, which include this relation, are not to be destroyed. It is advised that one never destroy "handwritten notes after rewriting or typing . . .you can always check back should you have doubts about the information in your typed file." It is further advised that these notes be kept separate from the reproduced notes. The explanation is that the source for information is a

primary record; notes taken from it, secondary; and typed copies are "third-hand."[15] The information in the data base and computer listings are also third-hand, not primary or secondary records, so the research notes must be kept. Also, the relation of the information to its source is not necessary in the collected fgw. The retained research notes keep information separate by sources. But the final evaluation puts it together without regard to source distinctions (although the sources are referenced). The fgw as used on-line, takes the place of the final, collected fgw, until the researcher chooses to conclude the research. Then he can print out final copies and write a report of the findings. The hand-written notes that are retained correspond to the separate source usage of the fgw. The collected family group information in the data base corresponds to the final, collective usage of the fgw. This resolves the issue of whether to keep a separate family record for each source of data, or to place everything in a single family record. The design will include one family group record per family.

The decision causes the replacement of a 1:1 dependency by a weaker 1:N. The relation between searches and families changes from a 1:N to an N:M. In the original system, an entry in the research log is cross-referenced to a set of fgw's through the search number, but never to more than one fgw per family. In the reverse direction, data on an fgw corresponds to a single search, and therefore, a single log entry. The relation between searches and families was 1:N, since it was 1:N in one direction and 1:1 in the other. But now, many log entries can apply to the same fgw. Each family record will be a composite record of several searches. The log-to-fgw dependency is still 1:N. But, in the opposite direction, one fgw is based on information found from a set of searches, each one with a log entry. The relation is now N:M, because it is 1:N in both directions.

The introduction of an N:M relation, caused by the decision to collect all fgw's for a family into a single record, involves the following trade-off. Retrieval of fgw's will be improved significantly, and a search problem avoided. In the original system, there is no need for a key for fgw's. All sheets for the same family are stored together. Finding a certain one, or all sheets, is just a matter of locating the family group and looking at each sheet in the vicinity. If the same method were used on-line, either all fgw's for a given family would have to be stored physically adjacent, violating physical data independence, or worse, the whole fgw record type would have to be searched sequentially every time any or all records for a family were to be retrieved. This is true since anything used as a family identifier is not unique in such an fgw file. The family identifier would return as many fgw's as there are applicable sources. In exchange for improved fgw retrievals, a search has become necessary when identifying the source of data on an fgw.

The elimination of the 1:1 dependency makes it impossible to find out where a particular data item on an fgw originated from, if using only the data base. It can be done using one's research notes. The data base does allow retrieval of all the sources for any fgw. This retrieval will return a set of search numbers that will limit the searching the user must do in his "off-line" file. The necessity of using the non-computerized file for this query is not a disadvantage of the data base. When the user wants to know the source of certain information, he usually wants to see it in the context of the other notes that were taken from the same source. There is no better place to look than his file of research notes. The value and use of the research notes as secondary records, as opposed to the third-hand computer records was stated previously. However, the notes consist of fgw's only. Neither pedigree charts nor research logs must also be maintained manually in "off-line" files.

4.2 Breaking the Record Form into Relations

4.2.1 Separating the Dependencies

A second problem to be resolved for the family group file, involves the handling of searches. Perhaps the domains associated with searches should be removed from the fgw, since none of the other fgw domains relates to the family represented in the N:M manner of the search-to-family relation. The search domains may also need to be separated further into document searches and source searches, if the distinction made in the research log is necessary for fgw's too. All the domains of the fgw are given in the Appendix. Below are some previously established dependencies and some new dependencies, all applicable to the fgw.

Document number → Date
Document number → Search number
Search number → Date
Document number → Document name,
  Document date, Place of Record,
  Type of Record
Search number → Place of Record,
  Type of Record
Father number, Mother number →→
  Child number
Father →→ Child
Mother →→ Child
Father, Child → Mother, Marriage ID
Mother, Child → Father, Marriage ID
Child → Father
Child → Mother
Child → Marriage ID of Parents

The first three have been established. The interpretation of the next two is that a document number refers to a single named dated document, which is located in a specific place, and is categorized according to the types of papers one possesses; a search concerns one of many types of documents or sources, located in a specific place. The rest of the dependencies are explained in the Appendix.

Much of this information is already included in the pedigree relations. The marriage ID domain added there is also added here. Since there is only one fgw

per family group, the unique marriage ID given to a husband-wife pair, is a primary key for fgw's. The marriage ID of the marriage relation is also the family identifier referred to in the discussion of the N:M relation. The binary relation between the marriage ID and the search domains is that same N:M relation. There may be multiple search numbers and document numbers for an fgw, given the previous decision to put all information for one family into one record. So, marriage ID →→ (search number, document number). Due to the search and document number dependencies listed above, the date of search, search number, document number, document name, document date, place of record and type of record can all have multiple values dependent on the marriage ID. Since marriage ID uniquely determines the husband, wife and a set of children, it is logical to separate the family domains and the search domains into different relations.

The family group record type will be split, leaving marriage ID, husband, wife and children information in the family group record type, and placing the other information elsewhere. The following discussion produces the additional relations split from the family group record type, and refines the separate document and source research logs produced from the original research log.

4.2.2 Placing Domains in Other Relations, Including a New Relation

Most of the research log domains are applicable whether the data was gathered from a source or a document. The only one that is not is the document number. The document research log contains a document number that cross-references to the document file described later, and that, along with date of search and search number, cross-references to the part of the record type split from the family group file. The relationship between document number, date and search number, doesn't have to be maintained on the fgw, since it occurs in the

document or source logs, as necessary. The association of document number to name is maintained in the document file of the next section. That record type could logically accommodate the domains, document date, record location and type of record, which are associations of the document entity. Thus, for a document, the removed family group domains are or can be maintained in the document research log or the document file. The document log now contains all the same domains as the original research log. Document number is the key, as described in the research log section. The document file obtains three new domains, document date, record location and type of record.

The source research log retains all domains except the document number from the original log. The same reasoning given in the research log section, pertaining to why a source doesn't need a related document number, applies here to why a source recorded on the fgw doesn't need a document name or date. The date of search and search number are associated with the source on the source research log through the source textual description. For a source, the only domains of those split from the family group file that remain to be accounted for, are place of record (source location) and type of record. These are associations of the entity set "source." They can be maintained in a new relation, source file, similar to document file (not to be confused with source research log and document research log). Source file domains include source name, source location and type of record. Source name should be a title, and possibly other descriptor fields like author and volume number, could be included to keep the, probably variable, source textual description - of the source log - as small as possible.

4.2.3 Retaining Information Through Creation of a New Relation

One more step remains in order not to lose information from the original fgw. The identifications of the individual sources and documents used to

compile the family record, have all been removed from it. Yet a common retrieval is to find all the searches that resulted in information on a particular family. Another is to find all the family groups that have information obtained from a given search. Also common are queries to find all documents or all sources that pertain to one family. The search number is related to a document or source. But a relation is needed to relate family groups with all searches applicable. A simple binary relation is marriage ID:search number. For any marriage (family group) there may be many searches that result in information for it; and any search may turn up information on several marriages. This has been discussed before in conjunction with this N:M relation. Both domains are necessary in the primary key. This relation could be called "research by family groups." It makes possible the queries listed in this paragraph. A search number from either the document or source research log maps to a set of marriage ID's, each a key to the family group file and to a marriage. Conversely, a marriage ID from a family group maps to a set of search numbers, each of which is a key to either the document or the source research log.

4.2.4 Refining Other Affected Relations

It was stated in the research log section that search number is a key in the source log. This is the case because every source has a unique search number. However, the surname domain may prevent this from being true when more than one surname is referenced in a source. This is probable. Either one log entry would have all the surnames listed with it, or a log entry would have to be repeated for each surname located. The reason for the surname field is to relate a search number to all the appropriate surnames. Yet the research by family groups relation connects a search number with not only surnames, but also specific family groups within those surnames. This goes a step further

than the surname : search number relation, making it unnecessary in the research log. (Some genealogists like to use the surname for a filing label. If the user wishes, the surname domain can be included in the family group relation, in addition to the individual-name domains.)

It was stated before that document number is a key in the document log. However, it is not the only key. Search number is a key also. The search number → document number dependency excluded in the original research log is valid for the research log that contains only documents.

The document log has two primary keys. The physical system chosen to implement it must allow one record type with two keys. The meaning of search number in the genealogical research log implies that every search number is unique. Thus the implementation must also ensure that search numbers for documents and those for sources are assigned from the same set. Any assigned search number may be a key in the document log or the source log, but not both. It is also possible that the search number isn't in either. The result field in a research log indicates whether any data was actually gathered and placed on fgw's. The reason for this is to record all searches, even if they are unproductive. Recording an unproductive search prevents one from repeating the same search months or years later. Also, someone who takes over another's research, will know that the document or source has already been evaluated, and won't waste time researching it again.[14] If the result domain contains the "no-information" value for a log record, then no information is available for that search in any other record types. This can avoid useless data base searching.

4.3 Handling Non-family Persons

Another fgw problem to be resolved is that not everybody belongs on an fgw. Some persons must be considered singularly. There are some who never had any spouse or children. There are some who may have had, but the researcher

doesn't have that information yet. Perhaps one does not know how a person is related to anyone else in the data base. There already exists another relation to handle this. The person relation obtained from the pedigree chart file is the appropriate relation. It was developed to keep names of individuals paired with the marriage of their parents, for either backward or forward lineage tracing. It contains much additional information on the individual. Every person represented in the data base should be placed in person, after being assigned a unique number. But not everyone need be placed in a family group. The parents' marriage ID of these isolated persons mentioned above may not be known, and would have to contain an appropriate null value. This would definitely be necessary for those people who represent the furthest back one's research has progressed: One simply doesn't have information on the parents or any other ancestors of these people.

The preceding discussion returns to the question raised in the pedigree charts section, concerning where to place personal data. There is still some overlapping information in person and family group. This is not necessarily inappropriate, and the redundant information should not all be removed. Genealogists are interested in each person, both as an individual and as a family member. Each person may be represented in two relations. In fact, as the design currently exists, each person can be represented in three places: as an individual in person, as a husband or wife in family group and as a child in family group.

4.4 Maintaining Children Information

The problem of how and whether to store children's information with the father and mother's, remains to be addressed. Maintaining a set of children seems like a linked list problem. This would require pointer maintenance (or

something similar) every time a child is born or "discovered." The method of fgw forms could be used, allocating a fixed maximum number of child domains to the family group record type. Usually some of these in each record would have to contain null values. Children information could be left out of the relation entirely. Data on each child is already kept in the person relation. It is retrievable through a search, as explained in the descendants-chart-processing discussion of the pedigree charts section. The marriage ID is used as a search key in person. Instead of no information, there could be one domain indicating how many children there are in the family. Or the indicator could be of the first born's unique number.

None of these options is ideal. The use of a linked list or a fixed number of "spaces" in a relation is not compatible with the idea or normalization of relations. There should be at least some indication that there are children in a relation that represents family groups. It is inconsistent to relate one child but not the others. Including only the first born in the family, is actually a form of using a fixed number of child domains. Also identifiers for the children can not remain in the relation. The reason is illustrated by the family dependencies for fgw's, listed in the Appendix, and earlier in this section. The identifiers for mother and any one of the children will determine the father. But the identifier of a child will by itself determine the father. The same is true for the determination of mother by father and child identifiers, and by child identifier alone. These are redundant dependencies, requiring separate relations. Therefore, a single domain is chosen to represent children information in family group. The values represent number of children, including values for no children, and number-unknown.

The linked list can't be totally dismissed. On fgw's, the order of listing the children is meaningful. The convention is to list in birth order.

To keep this order in the data base, a domain should be added to person for "sibling order." The first born would be given a value indicating first, etc. If unknown, the value could be "first," until later research placed this person elsewhere within a family. Or, there could be a null value. The use of null values in the sibling order domain, and in the marriage ID, as mentioned earlier for isolated ancestors, does not indicate a need for membership in a separate relation in these cases. A person's sibling order may not be known, but it is a value that everyone has. And a person's parents' identities may be unknown, but it is always the case that there were parents. The relations developed from the original family group record form are shown in Figure 2.5.

5. Document File

The genealogist's document file cannot itself be computerized. It is a manila folder or similar enclosure file containing certificates, personal legal and financial records, and other paper documents. The value of these is that they are primary or secondary records. Any computer recordings would only be secondary or tertiary, respectively. Computerization cannot replace the value of having original records. However, there should be in the data base, identifications of the documents in the document file, and indications of their contents. This is necessary for the complete cross-referencing and retrieval capabilities of the data base. Documents are uniquely identified by document numbers. A domain for document name is helpful upon retrieving information that certain data is contained in a specific document. Having the name in addition to the number of the original system, gives the user an idea of what the document is, so he doesn't always have to go to his enclosure file. The number and name domains would be enough to represent the document file on-line. But, development of the family group relations in the previous section showed

Figure 2.5 Resulting Family Group Relations

Family Group (Marriage ID, where married, number of children; father's name, when born, where born, when died, where died, when buried, where buried, occupation; mother's name, when born, where born, when died, where died, when buried, where buried, occupation)

Source (Search number[1], source name, source location, type of record)

Research by Family Groups (Marriage ID, Search Number)

1. Search number isn't necessary in the source relation, if source name is unique. But search number allows quicker retrievals.

a need for document date, record location and type of record domains. The key is document number. Document name is also a key if the user keeps them all unique. The relation is: Document (document number, document name, document date, record location, type of record).

CHAPTER III

DESIGN OF THE DATA BASE

1. Justification of the Model

With the human relationships involved, a hierarchy seems the most natural representation for the data base. Yet the data and retrievals presented in the previous chapter don't easily fit a hierarchical model. If a non-data base hierarchy such as a tree representation or other linked list is used, four or five links per record must be maintained. If they are not, some meaningful queries will be difficult or impossible to satisfy. Suppose a link exists from son to father, but not vice versa. Tracing lineage backward is simply a matter of following links. Tracing lineage forward consists of obtaining an identifier from a person's record, then searching all records at the next lower level to find any that contain a pointer to the obtained identifier; and then for each match do a similar search on the next lower level. The number of such searches per level increases at each level traversed. One of these searches is performed for each ancestor included, from the start of the trace to the next-to-last generation traced. The links that should be maintained for easy traversal are child, father and/or mother, spouse and sibling.

In a hierarchical data base, no links are allowed from a record type back to the same record type. These are mappings from and to the same or equivalent domains. This is unfortunate because this is exactly what is needed: a child, parent, spouse or sibling link is a mapping from and to the domain of people. E.F. Codd reported in his first paper on the relational model, "It is a remarkable fact that several existing information systems (chiefly those based on tree-structured files) fail to provide data representations for relations which have two or more identical domains."[8] The situation has not been completely corrected.

There is a model that could possibly be represented by either a hierarchy or a network. Each generation can be viewed as a separate record type or entity set. Furthermore, the males and females of each generation can be a separate record type. Paths to children, parents and spouses are links between record types. Tracing lineage forward or backward is simple. However, as this data base is built and used, there will be many broken paths, due to incomplete information. If one has reason to believe certain persons were ancestors, but doesn't know how they were related to anyone else, they cannot be included because their "connections" are unknown. Anyone without a known birthdate must also be excluded from the data base, because it's unknown to which generation he belongs. The generation limits will be arbitrary, and there will be some whose birthdays are on the borderlines. Links for siblings won't be accommodated since they would still require mappings from and to the same record type. If a person has married someone "young enough to be his/her child," it won't be clear where to place the spouse's record. If placed in the next generation, the record may need to be accessed by the link usually used to access son or daughter, in this case. Also, the number of record types in a set type would have to be dynamic, to allow for each generation researched. On finding an appropriate model, Stephen Johnson wrote, "The tool builder's challenge is to make the theoretical insights apply to the 90 to 95% of the problem the model fits, without making the remaining 5 to 10% impossible to do at all."(13) The problems and omissions of the hierarchical model eliminated it from consideration for the data base design.

The record forms used to derive the relational design could also have produced a network design. However, a relational design was chosen because it will save effort and reduce complexity for both the interactive user and the application programmer. The commands in a relational data base system are

more straightforward, so it's quicker for the user and the programmer to state a request:

> "Conciseness is the lack of verbosity. It is a measure of the quantity of code an application programmer must write, or the amount of typing an interactive user must do in order to express a request in a given DSL. It is generally the case that requests expressed in relational DSLs tend to be more concise than the same request expressed in the (network) DML. . .The DBTG DML deals with one record at a time, while the relational DSLs operate with sets."(19)

Additionally, results are achieved with less work when using the higher level of expression afforded in a relational language:

> "Perhaps what most strongly differentiates various DSLs is the amount of work a user must do and the ease with which a given result is accomplished. In other words, it is a difference in language level and complexity. Lower level languages tend to be more procedural. To the extent that a language allows the user to specify what is to be done without specifying how it is to be accomplished, it is said to be less procedural. . .The relational DSL SEQUEL is clearly a higher level language than the DBTG DML."(19)

Relational languages are less complex, so queries and application programs are easier to write:

> "The more complex a (language), the greater becomes the task of writing queries or application programs. In addition, the more complex. . .the greater the opportunity for logical error."(19)

The same authors go on to say that relational DSLs are less complex than network DMLs. Network users have the disadvantage that they must know and understand the access paths. They must "understand currency indicators and when they change (for DBTG)," and must be "responsible for their correct use."(19) A relational design was selected over a network, for all of the above reasons.

2. Normalization

An attempt was made to use Bernstein's Algorithm 2 to synthesize third normal form relations from the set of dependencies in the original system.

This was done after the development in Chapter II was finished, to verify that relations similar to those of that chapter would result. For the research log and pedigree charts, similar relations began to emerge. However, for the research log and the family group worksheets, the algorithm could not be completely carried out due to the presence of multi-valued dependencies. Multi-valued dependency considerations are not addressed in the algorithm, so it cannot be used. It could be that the problem of the multi-valued dependencies in this data base design is a semantic one.

> "The treatment of FDs (functional dependencies) in this paper is a strictly syntactic one based on Armstrong's axioms. . .Third normal form is a strictly syntactic property that is governed by the algebra of FDs. . .We are not attacking the problem of how to judge the semantic validity of syntactic inferences. Semantic problems of this type are not well understood and seem to be more difficult than the syntactic problem of determining 3NF."(4)

The relations chosen are not in third normal form. All attributes are simple domains, so the relations are in first normal form. All the keys are non-redundant, so they are in second normal form. But the redundant information (especially between person and family group) in the data base design, keeps it from being in third normal form. This fact and that the multi-valued dependencies are not based on keys, eliminate it from being in fourth normal form. The presence of the large number of multi-valued dependencies in the original system is the reason the semantics of these dependencies were carefully studied before normalization was attempted. It was thought that an understanding of the real-world relationships would suggest ways to handle the multiple values. The results of the study of the real-world relationships were data base relations that are in second normal form.

In the following discussion, the selected relations are presented one at a time. As each relation is presented, all possible redundancies with previous relations are examined. The result is a set of relations in the third normal form.

1. DRL. (doc. no., search no., date, descr., locality, tim. per., result)

2. SRL. (search no., date, descr., locality, tim. per., result)

These two relations both have date, descr., locality, tim. per. and result, but they never overlap. They are disjoint because the search no. key has disjoint values in the two relations. Still, they are the same domains, so they should be kept in the same relation. They can be, if the document number : search number relation is moved to a separate relation.

3. S. (search no., source name, source location, type-rec.)

DRL. or SRL. locality is not the same domain as location in S. or D. (following). "Locality" is the place that a record historically applies or refers to. "Location" is the place the record is currently kept.

4. D. (doc. no., doc. name, doc. date, rec. location, type-rec.)

DRL. date and doc. date are not the same domain. The first is the date of starting the research on the document; the second, the date appearing on the document. Source location and rec. location have the same domain, as do S. type-rec. and D. type-rec. Although they are disjoint, they can be attached to the new research log relation, as the other common domains are after number 2, above. Relations 3. and 4. can be pared down by removing source/record location and type-rec. The three new relations are shown in Figure 3.1.

5. P. (un. no., name, m/f, birthdate, birthplace, death date, death place, occupation, parents' marr., sibling no.)

6. FG. (marr. ID., where married, no. of child., (father, mother : name, birthdate, birth place, death date, death place, date buried, place buried, occupation))

Within the inner set of parentheses, all but two domains are redundant. This can be remedied by keeping all of those domains, with date buried and place buried, in the person relation only. Person is chosen because it is the relation that represents every person in the data base.

6.a. FG. (marr. ID., where married, no. of child, father un. no., mother un. no.)

7. M. (marr. ID., male un. no., female un. no., marr. date)

FG. has been reduced in size considerably, and unique numbers have replaced the redundant names. The family group relation now looks very much like the marriage relation. The previous representation of a married couple in family group was not a key since the names were not unique; but the replacement of names with numbers causes the two domains to become a key. This is reasonable because the same two domains in marriage, form a key (as described in Chapter II).

The two relations can be joined, since they have the same two keys. The two new relations from 5, 6 and 7, are shown in Figure 3.1.

8. RFG. (marr. ID., search no.)

Relation 8. won't be changed. Its purpose is to connect the two sets of relations 1-4 and 5-7.

Third normal form (as shown in Figure 3.1) now contains six relations, somewhat rearranged from the eight that existed before normalizing into 3NF.

Several multi-valued dependencies exist, in the revised design, which prevent it from being in fourth normal form. In fourth normal form, such a dependency must be on a key or keys only, and consequently only one such set is allowed per relation. Date in the new research log, name in person, male identifier and female identifier, separately in the new marriage, and each name in the old family group relation are all the multi-valued dependencies in the design. None of them are keys in the relations the dependencies occur in.

Since date $\rightarrow\rightarrow$ search number, document number and therefore all other domains in the research log, as stated in the previous chapter, then it can be removed from the research log and placed in a two-domain relation with search number. Search number is chosen as the domain to relate date to because

Figure 3.1 Relations in Third Normal Form

1. Document (Document number, Search number, Document name, Document date)

2. Research Log (Search number, Description, Locality, Type of Record, Result)

3. Source (Search number, Source name)

4. Person (Unique number, Name, M/F, Birthdate, Birthplace, Death date, Death place, Burial date, Burial place, Occupation, Marriage ID of parents, Sibling Number)

5. Marriage (Marriage ID, Male number, Female number, Marriage date, Marriage place, Number of Children)

6. Research by Family Group (Marriage ID, Search number)

because information would be lost if it were instead related to document number. (Not all searches are over documents, so document searches are only a subset of all searches.) The key of the new relation is search number since it always gives a unique date. But the key should include date, so that the multi-valued dependency is on a key, for fourth normal form. So both domains must be the key, to be in fourth normal form, but one of those domains alone is also a key. The key is then redundant. This same situation occurs when putting person into fourth normal form, which is explained later.

Name was removed from the former family group relation to obtain 3NF, resolving that multi-valued dependency, but it is still a domain in person. Name could be removed from person and male and female identifiers could be removed from marriage. But the results would be unfortunate. Male and female identifiers are both a part of a key in marriage, but neither is a key alone. Fourth normal form would not allow these two domains to remain in this relation. But the purpose of having them both in this relation is to relate them to each other, and give each pair of them a unique identification, the other key in the relation. Putting the marriage relation into 4NF would cause a single retrieval for this naturally occurring relation to be replaced by two retrievals. Also, fourth normal form of person would cause problems. If name is removed from person, it must be associated with unique number in some other relation. A retrieval of a person's identification, without a name, would not be very useful. If name : unique number is separated from person to become a new relation, name $\twoheadrightarrow$ unique number would appear in this relation only. The relation is in 4NF if name is a key or part of a key. It can't be a key by itself, since name : unique number is 1:N; but both domains can be the key. However, unique number by itself is a key, since unique number : name is 1:1. Now the relation is not in 2NF, because the key is redundant.

Unique number is a key and participates in another key. These reasons lead to the conclusion that fourth normal form for the person and marriage relations is not an improvement.

The question arises whether the normalization of this section is warranted. The update properties of the data base would be improved. But the relations as developed in Chapter II, are the result of planning quick retrievals based on certain frequent queries. That set of relations should at least be maintained in the user's view. It is kept in this report as the logical schema selected. The underlying physical schema may take into account the normalization of this section, but the remainder of the report will assume that the physical schema is also based on the arguments and results of Chapter II. A few improvements from the normalization process will be incorporated. Occupation, burial date and burial place will be domains in person, but not family group. The marriage date will be moved from marriage to family group. The replacement of names in family group by unique numbers will be retained.

3. Relations and Dependencies

Eight relations were presented in Chapter II. From the original research log, document and source research logs were developed. The relation

Document Research Log (document number, search number, date, description, locality, time period, result)

has two primary keys. Date is the only non-prime attribute that determines other attributes. All other attributes are multi-valued dependent on date. The relation

Source Research Log (search number, date, description, locality, time period, result)

has one primary key. Date has the same dependencies as above. From the original pedigree charts came person and marriage. The relation

Person (unique number, name, m/f flag, birthdate, birth place, death date, death place, burial date, burial place, occupation, marriage ID of parents, sibling number)

has one primary key. All domains, except m/f, are multi-valued dependent on name. The unique number →→ occupation dependency shown in the pedigree chart section of the Appendix has been removed by restriction only. Although one person may have many occupations, it is not feasible to add another relation just to account for them all. Usually it requires much research just to discover what the main occupation was. Retrieval of all a person's occupations is not a common operation. The multi-valued dependency unique number →→ marriage date shown in the Appendix has been removed. That date and the marriage ID refer to different marriages. The relation person →→ spouses has been removed, but the relation person → mother/father pair is maintained. The fact that a person married more than once is now recorded in marriage, where the person is paired with each spouse in a separate entry. The relation

Marriage (marriage ID, male number, female number)

has two primary keys, as discussed in Chapter II, "Pedigree Charts." The male/female key also adds convenience to a retrieval of a single marriage or set of marriages. The user doesn't have to know the marriage identifier, but instead can use the name or number of either party. Either male or female number alone, always determines multi-valued sets of marriage ID's, marriage dates, marriage places and spouse identifiers. The original fgw was the basis for the relations family group, source and research by family groups. The relation

Family Group (marriage ID, marr. date, marr. place, number of child., father number, birthdate, birth place, death date, death place, mother number, birthdate, birth place, death date, death place)

has one primary key. There aren't any other dependencies. The source relation

Source (search number, source name, source location,type of record)

has two primary keys. The relation

Research by Family Groups (marriage ID, search number)

contains only a two-domain key, where each domain determines a multi-valued set for the other. The original document file was the basis for the relation

Document (doc. number, doc. name, doc. date, record location, type of record)

has two primary keys. Record location and type of record may actually be related 1:N or N:M, depending on the user's filing methods, i.e., if different types of documents are stored in different locations. But, in general, the relation of the two can't be determined.

4. User Interface

To design the user interface, I have assumed the presence of a terminal screen and printer, along with a data base management system to control and access the data base. This discussion is only meant as an overview of one possible user interface that could implement the design. The explanations of the components are written as if this user interface has actually been implemented. The interface "interprets" terminal input for the DBMS, and formats DBMS output for the screen. Terminal input is edited, and messages are returned if there are errors. The input is translated to DBMS commands. (The commands could be statements in a data sub-language, embedded in a procedural programming language.) Execution of the embedded DBMS commands, invoked at the terminal, results in DBMS access of the data base, and appropriate output. The output is formatted by the interface and sent to the screen. It can also be formatted appropriately and sent to the printer.

The programmer producing the interface and the user are assumed to be two different people. The user is in effect a data entry operator. The user

needn't have programming experience, and doesn't have to understand the DBMS. The user interface is a group of application programs, working through a set of menu screen formats. The data movement through the system is shown in Figure 3.2.a. This interface is on an even higher level than a relational DSL. Yet, it is simple. The user doesn't have to remember commands or worry about syntax. All one has to do is call up menus and fill in fields. Only a small, fixed set of application programs, doing a stable set of processing, is probably sufficient for user needs. But if additional processing is needed later, screens and applications are easier to add on than commands in a user-query language. (These commands are actually used but are invisible to the user. The user doesn't have to start with a system that has a lot more capability than he needs, which would be confusing.) Thus, the processing is expandable.

If the relations didn't have redundant data, new information could be added by a series of hierarchical routines and calls. The three levels of routines would handle (1) Family and Research Logs; (2) Person, Marriage, Source and Document; and (3) Unique Numbering, Marriage ID assignment and Research by Family Groups (see Figure 3.2.b.). To add a new person, a routine first calls a lower-level routine that checks if the person is already in the data base, and assigns a unique number if not. Then the first routine adds in all the information input about the person. To add a new marriage, a routine first makes sure that at least one of the people is already in the data base, and that this particular marriage is not. Then the routine calls a lower-level routine to assign a marriage ID. The first routine then adds in the information input about the marriage, and calls a routine to place the marriage in the research by family groups relation, along with any searches that had been done for this marriage/family. (There would probably not be more than one search, when a new marriage is added.) When a new family is added, a

routine on the top functional level first calls on these person-adding and marriage-adding routines, and then adds in the descriptive fields for the family. When a document is added to the user's collection, a document number is assigned, and other input placed in the tuple. If any research has been done on that document, the search number is entered into the research by family group relation with marriage ID's of all the families involved, by a lower-level routine. This information is obtained from the input screen. After new research has been done, the user's input causes a check to ensure that the log entry has not previously been entered, and a search number is assigned. Either a Document Log or a Source Log entry will be made, depending on if the user had input a document number. Any addition to the Source Log causes a new entry in the source relation too. A value input by the user indicating "held" or "non-held" also determines if the Document or Source Log should be accessed. The presence of a "non-held" indicator and a document number would produce an error message, because document numbers are reserved for possessed papers. An entry in Source can't be made until the appropriate entry in Source Log has been made, but a Document entry can be made before or after the Document Log entry for it was made. (Sources are unknown to the user before the research, but a user knows what documents he owns.) In case the Document entry is made after the Document Log entry, the document number input is checked to see if it exists in Document. If not, it is placed in the relation, along with other document information. The research log routine is at the top level, and calls either the source-adding or document-adding routines. Either one of them calls the routine to handle research by family groups, adding the search number for all families listed on the input screen. The routines that assign person numbers and marriage ID's, and the one that places entries in research by family groups, are at the bottom level, inaccessible directly by the user.

The source-adding routine, on the second level, is not accessible unless a log entry for research done on a source is being added. The document-adder is invocable by the user directly, and from the research log handler on the top level.

Since there is redundant information between the family group and person relations, routine calls can't be strictly hierarchical. When data is placed in either of these, the other one will have to be checked, and updated if necessary. This complexity has been allowed in exchange for quicker retrievals when the query is for families only, or persons only. So it must be the programmer's responsibility to ensure that additions or updates to a family or person are recorded in both relations. However, this should not be too great a responsibility since the only domains present in both are unique number, birthdate, birth place, death date, death place and marriage ID. The only redundancies between person and marriage are unique number and marriage ID, the respective keys. The only redundancy between family group and marriage is marriage ID, the key of both. So the structure of Figure 3.2.b would need to be modified by only a person-to-family call capability, but each of the three routines handling these relations would have to be aware of what the purpose of the call to it is. And still, these routines should each be allowed to call each of the other two.

Changing data can be done in the same routines that are used for data addition. One of the input fields on the screen indicates whether the data is new or a modification. A routine checks the field to determine which processing is to be done. Modification processing is simpler than for additions, because assigning a unique number or marriage ID can not be allowed during an update. These can only occur for data addition, and the numbers should never be reassigned. Updating log entries does not involve much processing because the

user usually enters all the information he will ever have for the entry during the data addition. The only reason for a change is if values were entered incorrectly. The only routines that could have almost as much processing for changes as for additions are the ones handling families, persons, marriages, and the research-to-family relation. This is because values for entries for these relations are often obtained a few at a time.

Along with the system of applications, there should be a system of formatted screens. One of the screens would be for queries. Standard queries, such as for information on a family or person, all marriages for a person, all the research that has been done on a family and others addressed in this report, are listed for selection. A blank line is included for a query not in the list. One program handles all the standard queries. Another scans a keyed, non-standard request, for key words (or a restricted command format), interpreting the request for the DBMS. The standard queries listed should at least include those on the key of a single relation, and those where keys are used to carry such a retrieval across relations. Two queries that must be included in the list are forward and backward lineage tracing to produce pedigree charts. They can be represented by "1. Descendants of _____ for _____ generations" and "2. Direct ancestors of _____ for _____ generations," on the screen. Selection of either of these results in a message screen to record the progress of the transaction and periodically request additional input. This is illustrated by Figure 2.2. Each of these transactions automatically causes a printed listing. Any query produces a result on an unformatted message screen.

Another screen in the system is for maintenance. The user can instigate a back-up or restoration, perhaps using a diskette. A separate application program is needed for this. It would be a simple file-to-file copy. To save

space on the back-up medium, the records could be compressed when unloaded and expanded when loaded. The user might occasionally want a formatted listing of the records being maintained, but this should be an option selectable on the maintenance screen.

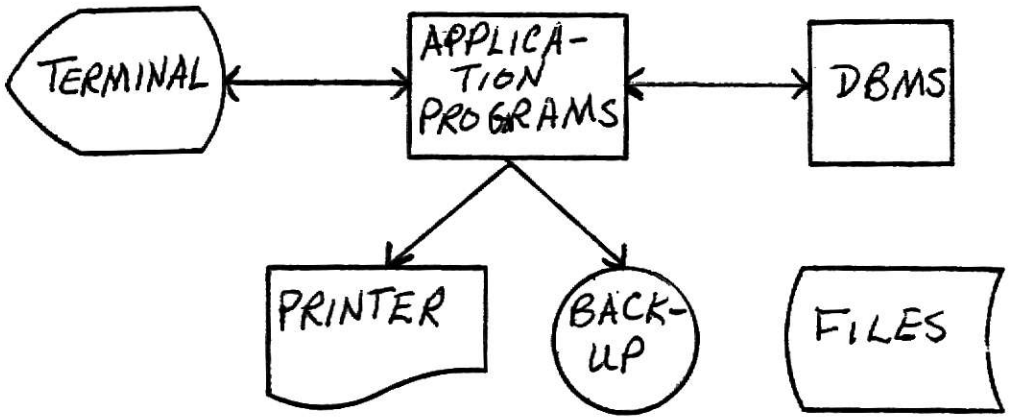
Ten screens (formats) are necessary in the system, one of them an unformatted transaction message/interaction screen. The initial screen is a menu of the kinds of processing available: listing help information, adding new data, updating, maintenance and queries. There is a screen for each of these. "New data" and "update" actually share a lower-level menu screen, where family, person, marriage or research screens are selectable from there. A field for selection of new or update can also be a part of this screen, allowing the user a second chance to determine whether the processing he wants to do is on new or existing data. The four screens at the next level (family, person, marriage, research) have a variable output field which says "new" or "update," depending on the selection made by the user on the previous or initial screen. Each of these four screens has fields for in/output of all the domains associated with the subject of the screen. When one of these screens is selected, all these fields (except "new/update") is blank. The user may enter a value in the key field or a secondary field, when updating. If a key was entered, all the field values for the record with that key would be returned. Then the user could update any values necessary. For a secondary key, all records meeting the criteria would have their keys returned, so the user could then enter a key. If new information is to be entered, the user can input all the values available. Then the transaction can return indication of success or failure. Necessary editing and checking would also be done. Selection of "help" on the initial screen would return a single screen (or pageable set of screens) of user documentation. The maintenance screen and query screen are also

selectable from the initial screen. Any queries from the query screen cause results to be returned on the unformatted screen, as previously explained. There are no other screens callable from the maintenance screen. The screen structure is summarized in Figure 3.2.c.

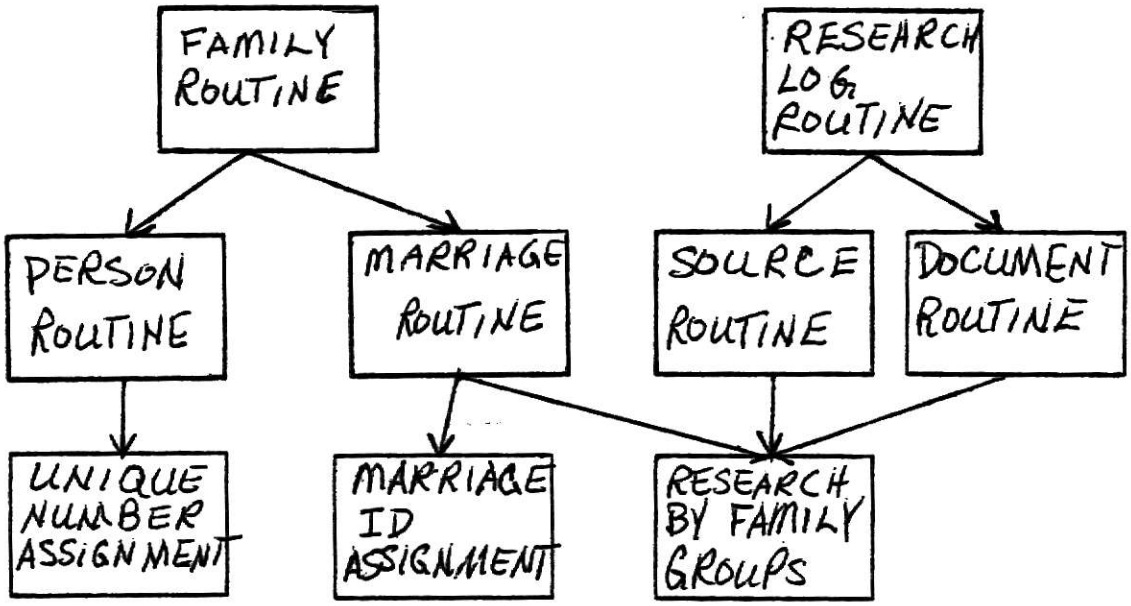
5. Implementation Considerations

The data base design must be implementable on a small machine - one with 16 kilobytes of random access memory and add-on memory of eight kilobytes. Home computers have these memory size limitations and limited access to secondary storage. The system produced from the design should be limited to using one disk or tape drive at any time. The use of a single drive complicates the back-up/restoration process and slows it down, because the files on one volume will have to be totally loaded into memory before they can be copied to another volume. The limited storage favors the relations of the design over 4NF relations. On a small machine, 4NF relations would cause more search delay. Since there is only one drive, all record types would either have to be stored on the same physical volume, or some of them would have to be loaded into the main memory. Loading them into memory every time one wanted to use the data base is not practical, because of the time it would require and the amount of storage the records would use that is needed for other purposes. Since all physical files for the relations will be on the same volume, a lot of time will be spent searching in order to access two relations during the same task, for example, when performing a DBMS relational operation. In 4NF, there are more relations and less information per relation, than in 2NF or 3NF. "More relations" means a retrieval may require access to more files than it otherwise would. "Less information" means a file located is less likely to contain all the data necessary to answer a query, so again more searching is required.

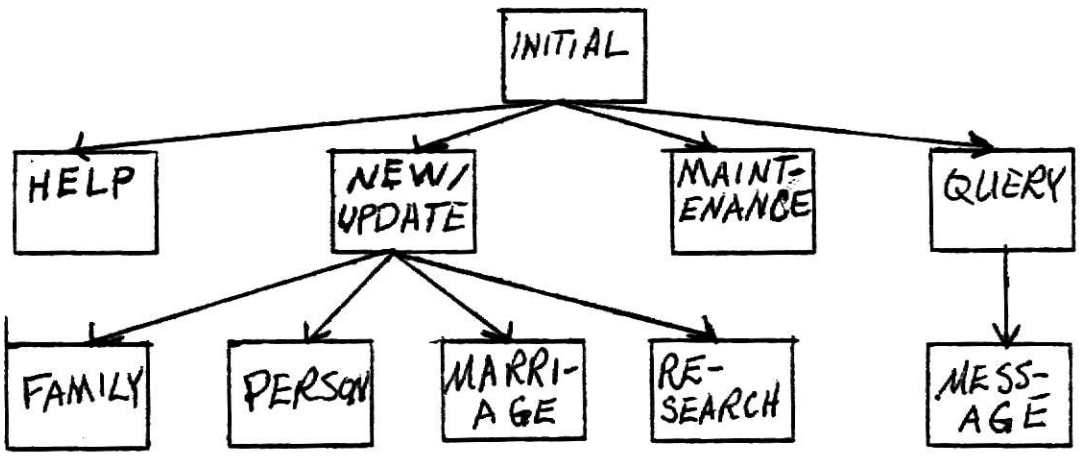
Figure 3.2 The User Interface



a. Communications View of the Hardware



b. Structure of the Functional Design



c. Structure of Screen Invocations

The data base design is implementable on a small computer. Storage for the data base, allowing room for data on 200 people, 100 families, 150 sources and 100 documents would require approximately 11 kilobytes. The source log relational tuples would require 2,200 bytes; the document log, 925. Person would require 4,200 bytes; family group, 2,300. Source would require 940 bytes; document, 425. Marriage would require 250 bytes; research by family groups, 200 bytes. All of these can be stored on a single diskette. Everything else can be kept on another diskette and loaded into the memory whenever the system is "started up." Besides the data base, the system needs room for a temporary file, the application programs and the screen formats. These are the files that would be loaded into memory. The temporary file would be used when a retrieval causes the construction of a temporary relation or when all or part of a physical file is being input or output. It should be as large as the largest relation, 4,200 bytes. That is more than large enough to use for two I/O buffers. Part of the area can also be used as a stack to allow recursion for pedigree chart processing. There are twelve routines, most of them very small. With an average size of 500 bytes, six kilobytes would be necessary. There are nine screen formats. Allowing 50 bytes for each, 1,350 bytes would be required to store them in a compressed form. In the main memory a total of just over 11 kilobytes is then necessary. Adding this total to the storage required for the relations, a little over 22 kilobytes will be sufficient.

CHAPTER IV

CONCLUSIONS

1. Results

Chapters I and II presented an introduction and analysis of a genealogical record-keeping system. Problems of retrieval, update, data addition and file growth were discussed in terms of both the original system and the conversion to a data base system. The record forms and files were converted to relational record types. The record types developed were based on the queries and retrievals peculiar to genealogical files. The on-line system designed is in general more restricted than the original. The cross-referencing in the off-line system had to be formalized and limited, to produce a design that was computer implementable. It was found that many multi-valued dependencies could be removed as a result of a study of the associations within the system, and of the way the system is used.

Chapter III presented the design of the data base system. An example user interface based on the design, was described. It was found that a relational model was the best for representing the system. The hierarchical model did not fit the data or the system. The network model was inappropriate for the user community. The relational model fit the data and the system. Also it would be more convenient for the application programmer and the terminal user. The relations developed were normalized into 3NF, and as far as practical into 4NF. The resulting relations were found to be less desirable than those before normalization, in terms of the system represented in the data base, the frequent queries and the hardware environment. An analysis of the approximate storage requirements of the system showed that it can be implemented on a small computer.

2. Future Research

Two assumptions that were made before undertaking this project are that the prices of home computer equipment will keep decreasing, and that there is a choice of DBMS's that are implementable on a small computer but won't consume all the available random access memory. The DBMS must also work on a non-distributed system, i.e., all its functions must be centralized. Such a DBMS is not yet available to the average consumer. There are several additional avenues of research to follow-up this report. A user interface based on the design can be developed. The data base system could be simulated on any computer system, or actually implemented on a large or small computer. An optimal implementation of the relations in 4NF could be developed so the access of the records would be simplified and speeded up. The design can also be extended to incorporate other record forms used by genealogists, producing more advanced systems. Other practical applications of the home computer can be investigated and designed.

The original goal of this project has been accomplished. A data base system for personal use by a genealogist, implementable on a small computer system, has been designed. A side effect was to show the practicality of a system of application programs used with a DBMS, for the at-home user, on a personal computer. Systems such as the one designed are a means to promote understanding and allay fears of the computer among the general public.

"The computer is one of those marvels of modern-day technology that instills many of us with trepidation and confusion. We know they exist, that they know all about our personal lives, and that they help the world think faster and operate more efficiently. But that's all we want to know about them.

"As with most of society's essential systems - plumbing, electrical power plants, modes of transportation - computers are considered necessary but too

esoteric for the average individual to deal with or to spend much time thinking about."[1] The increased availability and practicality of computer systems which meet the individual's needs will enhance the computer's acceptability, by demonstrating it merely as the tool it is.

REFERENCES

1. Apar, Bruce. "Computer Shock," Video, Vol. 4, No. 5, August, 1980.

2. Astrahan, M.M., Blasgen, M., Chamberlin, P., Gray, J., King, W., Lindsay, B., Lorie, R., Mehl, J., Price, T., Putzolu, G., Schkolnick, M., Selinger, P., Slutz, D., Strong, H., Tiberio, P., Traiger, I., Wade, B., and Yost, R. "System R: A Relational Data Base Management System," Computer, Vol. 12, No. 5, May, 1979.

3. Beard, Timothy F. and Demong, Denise. How to Find Your Family Roots, McGraw-Hill Book Company, New York, St. Louis, San Francisco, Dusseldorf, Mexico, Toronto, 1977.

4. Bernstein, P.A. "Synthesizing Third Normal Form Relations from Functional Dependencies," ACM Transactions on Database Systems, Vol. 1, No. 4, December, 1976.

5. Cardenas, Alfonso F. and Sagamang, James P. "Doubly-chained Tree Data Base Organization-Analysis and Design Strategies," The Computer Journal, Vol. 20, No. 1, Feb., 1977.

6. Chamberlin, Donald D. "Relational Data-Base Management Systems," Computing Surveys, Vol. 8, No. 1, March, 1976.

7. Champine, G.A. "Current Trends in Data Base Systems," Computer, Vol. 12, No. 5, May, 1979.

8. Codd, E.F. "A Relational Model of Data for Large Shared Data Banks," Communications of the ACM, Vol. 13, No. 6, June, 1970.

9. Elzinga, Mrs. Agnes. Personal Contact, Riley County Genealogical Society, Manhattan, Kansas 66502.

10. Everton Publishers, Inc., The. The Genealogical Helper, Vol. 35, No. 1, Jan./Feb., 1981.

11. _____ . Genealogical Supply Catalogue, The Everton Publishers, Logan, Utah, Jan., 1981.

12. Fagin, R. "Multivalued Dependencies and a New Normal Form for Relational Databases," ACM Transactions on Database Systems, Vol. 2, No. 3, Sept., 1977.

13. Johnson, Stephen C. "Language Development Tools on the Unix System," Computer, Vol. 13, No. 8, Aug., 1980.

14. Jones, Vincent L., Eakle, Arlene H. and Christsen, Mildred H. Family History for Fun and Profit (formerly The Jurisdictional Approach), Publishers Press, Salt Lake City, Utah, 1977.

15. Kelley, Harold H. In Search of Your Family Tree, St. Martin's Press, New York, New York, 1977.

16. Lichtman, Allan J. Your Family History, Random House, Inc., New York, New York, 1978.

17. Linder, Bill R. How to Trace Your Family History, Everest House, New York, New York, 1978.

18. Maynard, John. "The Open Channel: A User-Driven Approach to Better User Manuals," Computer, Vol. 12, No. 1, Jan., 1979.

19. Michaels, Ann S., Mittman, Benjamin and Carlson, C. Robert. "A Comparison of the Relational and CODASYL Approaches to Data-Base Management," Computing Surveys, Vol. 8, No. 1, March, 1976.

20. Schuyten, Peter J. "How a Personal Computer Changed My Life," Video Review, Vol. 1, No. 10, Jan., 1981.

21. Senko, M.E. "Data Structures and Data Accessing in Data Base Systems Past, Present, Future," IBM Systems Journal, Vol. 16, No. 3, May, 1977.

22. Schneiderman, Ben. "Human Factors Experiments in Designing Interactive Systems," Computer, Vol. 12, No. 12, Dec., 1979.

23. _____. "Improving the Human Factor Aspect of Data Base Interactions," ACM Transactions on Database Systems, Vol. 3, No. 4, Dec., 1978.

24. Sitz, Mrs. Golda. Personal Contact, Riley County Genealogical Society, Manhattan, Kansas 66502.

25. Taylor, Robert W. and Frank, Randall L. "CODASYL Data-Base Management Systems," Computing Surveys, Vol. 8, No. 1, March, 1976.

26. Tsichritzis, D.C. and Lochovsky, F.H. "Hierarchical Data-Base Management: A Survey," Computing Surveys, Vol. 8, No. 1, March, 1976.

27. Walker, A. and Wood, D. "Locally Balanced Binary Trees," The Computer Journal, Vol. 19, No. 4, Nov., 1976.

A P P E N D I X

1. Research Log

DOCUMENT NO.	DATE	SEARCH NO.	DESCRIPTION OF THE SOURCE	LOCALITY	TIME PERIOD	SURNAME OR NAME	RESULT

Dependencies:

Document number → → (Date, Search Number, Surname)

Document number : Description, Locality, Time Period, Result are indeterminate

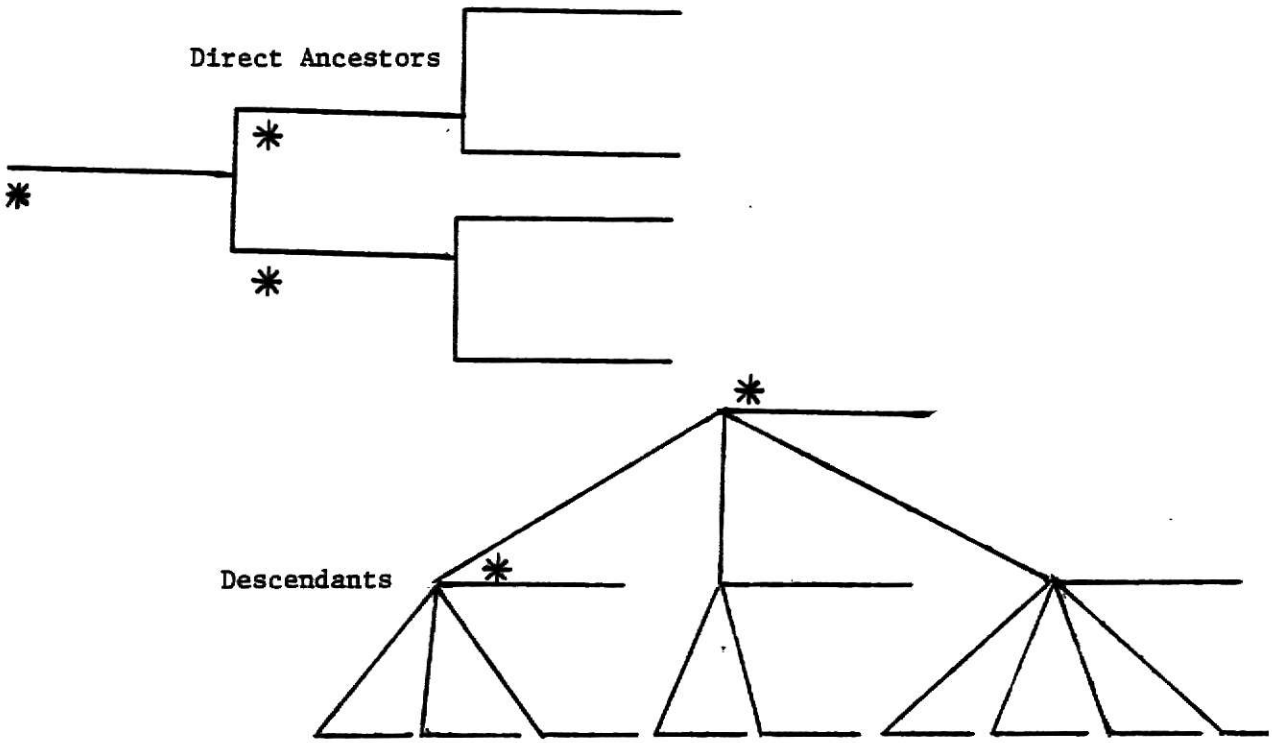
Search number → → (Document Number, Date, Surname)

Search number → (Description, Locality, Time Period, Result)

Date → →(Document Number, Search Number, Surname)

Date : Description, Locality, Time Period, Result are indeterminate

2. Pedigree Charts



* Fields for each individual are Name, Birthdate, Birthplace, Marriage Date, Death Date, Death Place, Number, Occupation

Dependencies:

Name →→(Birthdate, Birth Place, Marriage Date, Death Date, Death Place, Number, Occupation, Spouse, Parents)

Number →(Name, Birthdate, Birthplace, Death Date, Death Place)

Number →→(Marriage Date, Occupation, Spouse, Parents)

3. Family Group Worksheets

DATE OF SEARCH	SEARCH NO.	DOCUMENT NO.	DOC. NAME	DOC. DATE	PLACE OF RECORD	TYPE OF RECORD

Husband Name: _____ Wife Name: _____
Occupation: _____

When Where When Where

Born _____
Married _____
Died _____
Buried _____

Children: born died buried married to whom

Name: _____ when _____
where _____
Name: _____ when _____
where _____
Name: _____ when _____
where _____

Dependencies:

Document Number → (Document Name, Document Date, Place of Record, Type of Record)

All those listed in the Research Log.

(Husband, Wife) → → Child
Husband → → Child
Wife → → Child
(Husband, Child) → Wife
Child → Wife
(Wife, Child) → Husband
Child → Husband

Separate marriages for a person are recorded on separate sheets. Husband and wife together determine their children. Both are needed for the marriage represented because either one may have been married more than once, having children from each marriage. The husband alone determines all of his own children; and the wife, all of hers.

4. Cross-Referencing the Files

Document and research log are cross-referenced through document number. Document and family group worksheets are connected through document number also. The research log and family group worksheets are connected through search number, date of search and document number. Pedigree charts index the family group worksheets by surname and given name. However, multiple records will be retrieved. A name on a pedigree chart will correspond to (1) multiple fgw's for a particular marriage involving the named person, (2) for each of the multiple marriages possible, and (3) for several people who may have this name.

DESIGN OF A PERSONAL GENEALOGICAL DATA BASE SYSTEM

by

MARY JO BIRD

B. A., Fort Hays State University, 1976

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1981

ABSTRACT

The report presents the process of designing a genealogical data base system. The stages of development convert an off-line file system to a relational data base. The relations are derived from the record forms of the file system. The relationships are analyzed for meaning and usage. Various models for the data structure are examined, including family tree, linked list, hierarchical, network and relational.

The relations obtained are normalized into third normal form. A discussion of fourth normal form and some of the 3NF results leads to the conclusion that the unnormalized relations are more appropriate for the implementation constraints. The relations of the design are partly in 3NF, but contain redundant information. The implementation restrictions concern the probable user group and the properties of home computers, including memory size and limited access to secondary storage. A possible user interface is described for processing in the data base system.