Computers in Housing Authorities

by

Alicia L Allen

Submitted to the Department of Urban Studies and Planning
in partial fulfillment of the requirements for the degrees of

Bachelor of Science in Urban Studies and Planning

and

Master of City Planning

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1996

Author .....
Department of Urban Studies and Planning
Feb. 1, 1995

Certified by .....
Joseph Ferreira Jr.
Professor of Urban Planning and Operations Research
Thesis Supervisor

Accepted by .....
Mark Schuster
Chairman, Departmental Committee on Graduate Students

Computers in Housing Authorities

by

Alicia L Allen

Submitted to the Department of Urban Studies and Planning
on Feb. 1, 1995, in partial fulfillment of the
requirements for the degrees of
Bachelor of Science in Urban Studies and Planning
and
Master of City Planning

Abstract

Planners and other managers in housing authorities need to know what is happenning in their developments. They need to be aware of a multitude of facts regarding their tenants and the properties they own. They are constantly asking questions about the composition of their populations, locations of particular tenants, conditions of buildings and other organizational questions. These questions help them to make decisions about applying for grants, spending their funding and planning for the future.

This thesis discusses the issues involved in allowing a planner direct access to the data available at a housing authority. This access allows them to answer these questions themselves, without relying on someone such as the MIS department to do them for them. It also addresses the particular set of tools that a planner might use in organizing and in accessing data using relational database management and SQL.

After discussing these various issues and possibilities in data management, I present some questions that planners may wish to ask. These queries are divided into several types. These types include queries that can be answered by practically anyone using today's systems, and other questions that get increasingly more complex such as how many of our tenants are of which race? Has this picture changed over time? What are the patterns of gainfull employment among tenants? Do long and short term residents tend to live in different parts of the development? I discuss some difficulties that might arise in answering these questions, and some possible solutions. Finally, I use a relational database and SQL to answer some of these queries. This section presents both the simplicity in answering some of the questions and the obstacles and complexities involved in others of these questions. In a final section I present maps that one might create to illustrate some of the questions answered in the previous section.

My conclusions discusses the benefits of and obstacles to giving a planner direct access to data in his or her housing agency. These include issues of data security and complexity as well as the benefits of saving time and resources. With direct access

to the data a planner can follow her own line of thought and questioning without repeated interactions with the MIS department. I also mention some of the things we can anticipate in the future of computerization and data management. This section particularly mentions on-line communication and data sharing both within and among agencies.

Thesis Supervisor: Joseph Ferreira Jr.
Title: Professor of Urban Planning and Operations Research

Acknowledgments

There are alot of people I'd like to acknowledge, specifically my advisor, Joe Ferriera for putting up with me all these years - for finding research around the lab for me to do when I needed a job and for not minding when I ended up doing other work. For Lang Keyes for the many classes of his I enjoyed, even if they were at 8AM and for reading my thesis for me. I'd like to thank the many people who were totally understanding of moods when my thesis wasn't going well, particularly those who were at dinner after my first defense. I'd like to thank Kathy Peck for proofreading my thesis and Katy Oldham and Gary Dryfoos for offerring. I'd like to thank the Spring 1995 11.520 class for its work on Mission Main, and giving me the opportunity to explore the questions planners there are asking. I'd like to thank the employees at Mission Main for their kindness in answering my many questions and providing me with lots of information. I'd like to thank the various people at the many software agencies I called, for their time and interest in my project. I'd like to thank the makers of Oracle and ArcView for their wonderful products that made my thesis so much easier to write and illustrate, and particularly Microsoft for those wonderful programs Word 6.0 and Excell 5.0.

I'd like to thank my roommate for moving out, thereby giving me an environment more conducive to working, and all the people I zephyred who helped to keep me sane. Particularly those late at night.

I'd like to thank Sandy Wellford for all her help in making sure that I have enough units in the right areas, and all those other administrative details that needed to happen so that I could graduate.

And what thesis is ever done without thanking wchuang, William Chuang, for his help through olc in making latex 'do the right thing' (at the last moment, of course!). This time I get to thank his girlfriend, joanna, Joanna Yun, for getting a hold of him and making him do it, as well.

Table of Contents

Chapter One Introduction

Introduction

Planners and other managers in housing authorities need to know what is going on in their developments and about the people they are assisting. They need to be aware of a multitude of facts regarding their tenants and the properties they own. Planners need to have information about who they give their subsidies to, where those people live, information about these people and information about these properties. Planners constantly need to know information about numbers tenants and number of tenants in different categories, such as race and age. They are constantly asking questions about composition of their populations , locations of particular tenants, conditions of buildings and other organizational questions. These questions help them to make decisions about applying for grants, spending their funding and planning for the future.

This thesis discusses the issues involved in allowing a planner direct access to the data available at a housing authority so that they are able to answer these questions, and the particular set of tools that a planner might use in accessing these data. Tools such as a relational database and SQL can be used to answer these questions and organize the data in a clear manner. In this thesis I mention some of the other tools that can be used for these purposes, but for the most part I focus on these tools for data management.

Many housing authorities have realized that this is a good time to invest in their information systems and how they organize their data. For some this means starting from scratch, and for others this involves updating existing systems. When most agencies think of their computer systems, they regard them in administrative terms, i.e., tools to keep track of their finances, tenant lists, Federal reports and other bookkeeping needs. Less often, they view their information processing as an integral part of planning, and view their computer systems as an important planning tool. This thesis examines the feasibility of allowing planners direct access to data. This examination looks at software for housing authorities, in particular relational databases as a tool for facilitating planning. To do this, I suggest queries that would be of interest to the Boston Housing Authority with respect to its Mission Hill Urban Revitalization Project. I take an in-depth look at the BHA data and answer the queries using these data. I also review some of the software developed specifically for housing agencies, that allow planners and other staff members more direct access to the data.

In order to consider giving planners direct access to the data, it is necessary to consider the fact that some housing authorities may not be automated at all, which brings in additional considerations. The issues involved in bringing an entire housing authority on-line must be considered, in addition to the issues involved in giving multiple people direct access to the data. Until recently, when people thought of installing a computer system, they automatically thought of automation: "Can I keep track of who lives in which unit, with a computer, instead of by hand?" Often people only expect a computer systems to allow them to continue operating as they always have. However, this attitude obscures the vast opportunities for operational change that can be facilitated by a new computer system. Today, people on the forefront of technology rarely speak of "automating". Instead, they refer to "installing an information system." This is because automating current tasks is only one of the many functions that computers are now able to perform. Computers today do more than store data; they also allow users to display and manipulate the data as they wish, which can greatly enhance the value of the data. Information systems are also a great deal more than just computers. An information system may include networks of computers, printers, fax machines and even telephones, as well as the organizations and skills of those who use them.

I will use the term "information system" to refer to this wide range of communication tools. I will use the term "computer system" to refer to only the computer parts of such a system, such as workstations, servers, personal computers and their operating systems and software.

There is presently a great deal of flexibility in computer systems that did not exist ten, or even five years ago. Today's systems allow users to do more than simply enter and retrieve data. Users can now organize data more effectively, and new technology enables more accurate extrapolation and prediction. Computer systems are now able to support ad hoc data needs, which are defined on-the-fly by end-users.

The abilities to make ad hoc queries and to interact with the data on-line are beneficial not only in housing authorities, but in many fields. Agencies, companies and businesses in many areas are upgrading and installing computer systems. Many of the issues that arise when installing a new computer system are similar across fields, and I will be looking at discussions of reengineering and installation of new computer systems in the area of small businesses. I will draw these discussions into the housing context where they overlap.

Housing agencies range in size from the very small neighborhood nonprofit to the city wide housing authority. This difference in size is significant in many ways. Larger agencies tend to have larger staffs and budgets, which allows them to allocate more resources to information systems. Larger agencies might also have many offices at different locations, thereby increasing the complexity of the communication system. Most importantly, larger agencies have more data spread among more people and locations, which can make the task of storing and analyzing it considerably more complicated. In a small agency that supervises less than 200 housing units or certificates, it is reasonable that there might not be a Management Information Systems (MIS) department, that all the data are kept on a single computer and that the computer is maintained by a single person. With such a small operation, it would not be difficult to use spreadsheet tools to store, retrieve, update and analyze the data, and storage space would not be an issue. However, as agencies get larger, the issues of how to best view the data and how to keep storage space manageable become very important.

This thesis is based on ongoing research I have performed with regards to various housing agencies. I worked as an undergraduate research assistant mapping the locations of housing for one of the local housing agencies. This entailed getting data from the agency, and formatting it to work with the computer systems at MIT, as well as georeferencing the data. I then worked as assistant to the MIS Director at this medium sized housing agency. During this time, they were preparing to install a new software package to manage the tenant lists and inspections in their agency. I also worked with a class at MIT while they were acting as computer consultants to the Boston Housing Authority (BHA). The class concentrated its efforts on the data related to the authority's Urban Revitalization Project (URD): a $50 million, 5 year project to revitalize one of the city's housing developments. The students used their computer skills to assist the planners and to establish some baseline measurements to evaluate the progress of the revitalization over the five years. Through this project I was able to examine how the agency stores its data, and to witness the types of questions planners in a housing agency might want to ask of the data.

In this thesis, I first look at some of the literature in the areas of the automation of housing agencies and small businesses, and issues in planning where data analysis can be helpful. I then look at some of the software packages that have been developed particularly

for housing agencies, in order to see what database-related planning tools the market offers today. The next section of my thesis discusses the type of queries that planners might want to ask when working in housing agencies. It examines what sort of data and tools are needed to answer these questions, and what makes answering these questions difficult. I then use SQL and an Oracle database manager to answer some of these questions and a GIS package, ArcView to map some of the results of these queries. In my conclusion I reflect on the feasibility of direct access to data for planners in housing agencies, as well as where we can go from here, and what the future holds for computing in housing agencies.

I begin here with an introduction to housing authorities. The intent of this section is to give the reader enough background about the functioning of housing agencies so that he or she will understand how the planning and computer issues fit in later.

Housing authorities maintain public housing. Public housing consists of housing units that are owned by the government, where low-income people pay rent based on their income. When public housing is grouped together in very large buildings, all owned by the government, it is often referred to as "Projects," or the current politically correct term, "Developments." The housing authority administers the waiting lists, assigns low-income people to apartments and then manages and maintains the buildings. From time to time the regulations about income and rent change; currently residents must have a low enough income to get a unit in public housing, and then they pay approximately 30% of their income as rent. There is a significant amount of paperwork involved in ascertaining that an applicant for public housing is eligible. Then, once the applicant receives housing the authority must keep track of the size of the household, the household income, and unit characteristics, in order to know what rent to charge. The accounting for public housing can get complicated since each household pays a different amount of rent. Unlike in the open market, the size of the apartment a household occupies does not depend on the rent they pay, but rather on the size of the household. There are regulations about how many bedrooms are needed for different size families, based on the sexes and ages of the members of the household. Therefore, this information must also be maintained by the housing authority. As the family composition changes, and the ages of the household members change, the household may have to move from apartment to apartment within public housing. According to government regulations, the housing authority also keeps

track of the race and ethnicity's of the family members, as well as information regarding disabilities and other income sources. All of this information can be very complicated to maintain and keep up to date.

Housing authorities may choose to track other information about their tenants for their own uses as well. This may include information regarding employers, the dates households move in and out of apartments, and reasons for moving. The information can become very complicated to store, since individuals may have several employers or sources of income, and households may move in and out of several apartments during their stay in public housing.

The authority also needs to keep track of information about the properties they own. They must maintain information about the condition of the housing, its location and its size. Information is also often kept about who moves in and out of each apartment and when.

Housing authorities may also administer various voucher and certificate programs, as the government implements them. These programs require the housing agency to maintain much of the same information as for public housing, as well as additional information about where the household is living, who the landlord is, the condition of the apartment and the amount of rent that the housing authority needs to pay the landlord on top of what the tenant pays. The authority also needs to keep track of how many certificates or vouchers they have for each program, and the various regulations that go along with each program.

The most common types of certificate programs are the federal "Section 8," and various state programs such as the Massachusetts "MRVP" program. This money is distributed as certificates and vouchers. A voucher or certificate represents a certain amount of money that the government provides for housing. The rent the recipient pays must be less than, or the same amount as a certificate. In the case of vouchers, the recipient must pay the difference, if the rent on the apartment is more than the voucher.

When government subsidy pays for the apartment, the apartment must meet various safety standards. The agency that distributes the vouchers is responsible for guaranteeing that the housing meets these standards, and for keeping careful track of this information.

Housing authorities are also bound by many regulations regarding the housing they own, written by both the federal and state governments. These regulations cover things

such as minimum standards for buildings in fire and safety codes, as well as less obvious regulations and restrictions. These regulations and codes are often much stricter than general health codes. The housing authorities must produce regular reports regarding all of this information, so improved computer and database systems can ease the burden of these reports.

Nonprofit housing agencies need to maintain much of the same data needed by public housing authorities. These agencies sometimes own and maintain their own housing, for which they can make their own rules and regulations. However they also often distribute Federal housing funds from HUD, the Department of Housing and Urban Development. When they distribute these funds, they have the same requirements for record keeping imposed on them by the Federal government that public authorities have.

The complexity of the data needs of housing authorities are even further increased by the fact that they need to keep careful track of their finances, often in much greater detail than other businesses. When a housing authority receives a grant from the government, it is usually for something very specific. For instance, they might receive money to fix the roofs of the buildings, and even if the front steps really need to be redone, the money can only be used for roofs. Historically, most money that a housing authority receives is for physical improvements on existing buildings or for rent subsidies. They have received almost no money for social or community programs. As a result, public housing authorities have rarely dealt in such programs. However, the Federal government is testing a program of comprehensive development, forcing agencies with no experience in social services to offer such services. This program is discussed further in the section on the Boston Housing Authority.

Nonprofit agencies tend to have slightly more flexible funding sources. As a result, they are able to obtain money for social service programs. This money also tends to be fairly restrictive, such as "funding for after school programs for 3rd & 4th graders" or "funding for a program for teen-mothers with one child so they can obtain their GED." As a result of this inflexibility, integrating housing subsidies into a more comprehensive strategy of assisting poor families and providing self-help assistance in volume creates complex accounting and record keeping.

The process by which a household obtains a spot in public housing or receives a voucher is also complicated. To apply for a housing subsidy, a person first comes to the

agency and fills out an application. This form is then checked to see that the proper information is filled out, and that the applicant meets all the requirements. Often more data are needed, or the person is not eligible because he or she is over the income limit, or because the apartment where he or she is living is overcrowded.[1] Once the form is properly completed, the recipient must come in for an interview, bringing with him or her documentation of all the information provided on the form. This information is later entered into the computer by another employee, and the applicant is placed on a waiting list. As vouchers or units become available, recipients are notified and are moved from the waiting list to the tenant files. At most authorities, they are assigned a case manager, and then begin to receive government funding. Every year the client must attend an interview with the case manager and the apartment must be inspected. Records of all this must be kept and reported annually to HUD. If the agency receives funding from other sources, it is often necessary to send additional reports to those other sources as well.

After agencies receive money from HUD, they distribute most of this money, as rent, to the respective landlords. This effort is the part that is most often already automated. Most agencies prefer to be able to do such things as send one check per landlord rather than one check per tenant, since landlords with government subsidized tenants often have more than one subsidized tenant. It is also useful to be able to stop checks to a landlord automatically if there is difficulty with an apartment.

The maintenance and upkeep of these data alone can be a huge and complex task. The complexity of this required task has traditionally led housing authorities to simply keep the data and report on it. It has been rare for a housing authority to examine the data or use it for planning purposes. When there was a need for more housing, the authority would build more if it could raise the money. When new government funding becomes available, housing authorities channel the money to the appropriate places. There is often very little significant planning or research within the public housing authorities. This is generally due to the already huge task of maintaining and managing the existing properties, and the information about them. With the advent of less expensive and more powerful computers, this need no longer be true. Over the past 30 years, databases have slowly been evolving to the point where complex, on-the-fly queries are within reach of everyone: Mace described it as "information mining" (Mace 1994). New technologies are allowing the evolution of

[1] The Federal government will not subsidize overcrowded housing. Therefore the applicant must find an

faster and larger databases. Now that it is possible to keep extensive amounts of information on-line, and adequate tools for searching and organizing data have been developed, planners can begin to formulate and ask the questions that allow them to function more effectively.

Not only do new technologies give us greater access to information, but these technologies also allow us to view this information in a more timely fashion. In the past housing agencies have acted very linearly. The authorities have waiting lists and they would like to find ways to house everyone on their lists. They also realize that there are problems in their developments, and they would like to improve them. Housing agencies have always realized the importance of improving the security of their buildings and making the infrastructure of their buildings such that it is able to withstand a great deal of abuse. To accomplish this, public agencies apply for grants and make budget requests to the government, while private agencies apply for grants and look for loans and other funding that they can leverage from the private sector. When agencies can afford to, they build housing or make repairs to their current housing stock. There is very little research or analysis involved. There are several reasons for this: One reason for lack of planning is that the problems of housing agencies are immediate, and cannot wait for reports and analysis that might take months to generate. Surveys and research take time and human resources, both of which cost the agencies money. Often agencies are given money that needs to be spent by a certain deadline, or the money will disappear. In cases like this, the agency does not have the time to allot to research: they must act immediately. Sometimes, the needs of their current residents are so pressing that they must fix up those buildings immediately, rather then do surveys and analysis of what might be the best way to improve all of the residences. Because of these immediacies, housing authorities have often not done the best job possible, but rather the most obvious and immediate job possible.

The advent of less expensive, more powerful computers has made possible some of the research and analysis that these agencies would like to perform. Because of the growing ability to store more time sensitive information, and the increased capability for relational analysis, it is now possible for housing agencies to do useful analysis and research in a fraction of the time, using a fraction of the resources that it used to take. Housing agencies used to keep all their records on paper. Paper takes up a lot of space, deteriorates over time,

apartment that fits their family size before they are eligible for the subsidy.

and is very unwieldy to sort through and analyze. When agencies first started using computers, they were only able to keep some of their data on the computers. Due to the high costs of computer storage, they were forced to off-load their data to backup tapes or hardcopies periodically. This made it difficult to access data from even the recent past. Often data as recent as a year or six months ago would no longer be readily available because of the cost of storage space. Due to recent advances in the computer industry, having this data available is much less expensive today. It is now possible to keep thousands of kilobytes of data around, readily accesible for query and analysis.

Immediate access to the database can be useful throughout the housing agency. When a tenant calls the housing agency, the person who speaks with them must find their file in a file cabinet, in order to have that tenants information in front of them. Desktop systems allow the housing agency employees simply to enter the tenant's name into the database on a desktop computer, so that all of the tenant's information appears automatically. There is no time spent searching through files, and no risk that a piece of paper will fall out and get lost. All of the information is at the employee's fingertips. The employee can also update the information in the computer directly when the tenant calls or comes in for an interview. In most agencies today, when a tenant calls or comes in to update personal information, the staff member who takes the information writes it down on a form which is put in the file, and then on another form which is sent to the data-processing department where the information is entered into the computer. This system results in redundancy and delays, and is prone to introducing errors into the data. Copying the update onto a form and sending it to data processing to be entered often results in several delays, and has many opportunities for the data to be miscopied. If the employee who got the information from the tenant was able to enter the data herself, it would be entered in a more timely manner. It is also less likely that errors would be made, because she would only have to read her own handwriting when entering the data, and would not need to decipher someone else's handwriting. Direct access when updating the data is more efficient and reliable.

Another area where having direct access to distributed information would be very beneficial is on the housing site. When a housing authority has developments of several buildings, they usually have an on site office and a site manager. For this manager to get any official information about her clients, even to find out what apartment someone lives in, she must call the main office, or send a form to the office, requesting the information.

Often it takes so long for reports to be gotten from the main office that it is not worth the wait. With a distributed system, the site manager could access data directly from on site when she needs it, and in the form she desires. A distributed system where people can access the data from anywhere within the agency allows the agency to work more efficiently, and helps insure the data's integrity.

Of course, such real time, decentralized computing systems require a level of coordination and control that is more technically complex than traditional accounting systems. For example, since multiple users can access data at the same time, protections must be in place to insure that changes to data records are correctly updated. Also, upgrading distributed computing systems as hardware and software change, can be a complex undertaking. A goal of this thesis is to assess the complexities and benefits involved in making administrative housing authority data more readily usable for planning purposes.

While computers play an important role in the organizational aspects of housing authorities, databases may also be used to address strategic planning issues such as those addressed in the BHA URD project, and discussed later in this thesis.

Chapter Two Background

In this section, I will present background on the Boston Housing Authority. I then review some of the literature regarding housing agencies and planning support tools. In the last part of this section I discuss some of the software that is written specifically for housing agencies.

Boston Housing Authority Background

The Boston Housing Authority (BHA) is considered a large public housing authority. It maintains 12,500 public housing units on several sites scattered around the city. It also manages approximately 10,000 certificates and vouchers. The BHA estimates that in the city of Boston approximately 60,000 units are subsidized housing, accounting for one-tenth of the population. This is one of the highest concentrations of public housing in the country.(BHA interviews 1995)

In 1982 the BHA upgraded their tracking system from a manual card index system to a McDonnell Douglas Series 9200 computer with the "Public Housing Management Software" package from Creative Computer Solutions running on it . (Goodrich 1989). In 1994 they upgraded this system to a hierarchical PICK Unidata SUN, still with software from Creative Computer Solutions on it, but with increased functionality.

My interaction with the BHA involved working with the Mission Main Urban Revitalization Demonstration Project (URD). The BHA has been granted $50 million from HUD over a 5 year period to revitalize the Mission Main housing development. The strategies behind this project are that the Authority is given autonomy to make tradeoffs between physical, security and service improvements and it is allowed to ask for waivers from federal government regulations. Hence, the program recognizes that housing developments are intrinsically related to the areas surrounding them and that their successes depend on integration of the developments into their surroundings. This will be the first time that the BHA is working with social service agencies on any large scale.

The development itself is 26 acres, with 822 units, in 38 buildings. Five hundred fifty-five of the units were occupied when the project began. There will be 538 occupied units, in 34 buildings, after the renovations. Most of the tenants are leaving by receiving Section 8 vouchers, while a few have been evicted for cause (BHA interviews 1995).

The development is in a very desirable location: It is in a middle class neighborhood, between two subway lines (orange and green). There are many hospitals and several educational facilities in the immediate vicinity. The Longwood Medical Center is directly across the street, and has already indicated an interest in working with the development.

Mission Main was once known as the "largest open-air heroin market in North America." In September of 1994, a huge raid to clean out all of the drug users and pushers was very successful. Other steps that are being taken to improve security are community policing in the development, and reconstruction of the buildings with an eye towards safety. It is hoped that raising the minimum income needed to live in the development to 80% of the median income will also attract more working class people.

My interaction with the BHA is through this project and a geographic information systems class for which I was the teaching assistant. The class's objective was to provide baselines against which to measure the success of the URD over the next five years, as well as to provide information to the BHA that would assist them with the project. This project gave me access to the data at the BHA, and provided me with insight into the types of questions that help planners in a housing agency. We began by interviewing the director of the project, the person in charge of social services for the project, and the director of MIS at the BHA. We then went on a site tour where we were able to ask questions of the development's manager and to see what the surrounding neighborhood is like. Further steps involved getting actual data from the BHA and storing it in a format that would facilitate our research. How we dealt with the data helped us to recognize what makes answering planners' questions more or less difficult. In some cases the BHA simply did not have the tools or resources to analyze the data, and in others, the way the data were stored did not lend itself to analysis. This will be discussed in greater detail in my chapter "Difficulties of Direct Access".

The BHA has a department of Management of Information Systems (MIS). This department oversees all the computer operations of the agency. All reports or address lists needed by professional staff are ordered from the MIS department. The department has recently installed a prepackaged piece of software from C.C.S. (Creative Computer Solutions) which runs on a SUN 1000. The database is a PICK Unidata database like Prime Info. They have moved their data from a McDonnell system. Within 36 months they would like to have PC's on site at all their developments, so that the site managers can

retrieve data from the system without going through MIS. The communication network will be composed of either Frame relay or ISDN. They are upgrading the phone system and want to piggyback the digital system on it. This system will allow the people at the developments to access information from the main database themselves. Site managers will have access to information such as whether a work order has been entered, whether it has been filled, or whether a tenant's rent check been entered into the system yet. Throughout the agency they are now using the Lotus suite of tools, Word Perfect, Lotus 123, and will soon install Notes for daily communications.

Automation of Housing Authorities

In this section I examine some of the literature that exists in the areas of automating housing authorities and the issues in housing agencies today that make automating necessary. I also discuss some of the skills and knowledge that are needed by planners to make effective use of the tools available. I start with a discussion of issues faced by housing agencies that are automating, from the *Journal of Housing*.

There are many difficulties involved with the implementation of a new computer system. Some of the issues involved are data conversion, support of management and support of staff. These issues are discussed in several articles in the *Journal of Housing* and in the literature on office automation and MIS.

One of the first issues that must be confronted is management support. This is an important factor in a successful implementation for several reasons: Management provides funding and allows employees time to learn the new system. Without management support, implementing a new computer system is virtually impossible (Beam, 1990; Rockart).

In many businesses and companies, when reengineering computer systems, great emphasis is put on senior management and what information they would want from a new system. However, until recently, senior management has often had little or no experience with the system and therefore have had little interest in it, other than asking "does it work?" A great deal of research based on interviews with CEOs and other top managers has focused on what information they need to have available for future planning (Rockart lectures). Such information should be utilized in designing or choosing an information system in a housing agency as well. This aspect of computer use is illustrated in the discussion of questions planners might want to ask.

Staff support is important as well, because they are expected to use the new system. It is often useful to give front-line staff access to the data. Examples of this include saving a great deal of time and effort by having the person who deals with a client enter the data regarding that client directly into the system themselves, possibly even while working with the client. This reduces both human error and the redundancy of needing paper forms, and also assures that data are entered in a timely fashion. By allowing site managers access to the agency's data, you greatly reduce the need for reports from the central office. This also allows managers to get additional information that they may have previously felt not worth the trouble, and it allows them more timely access to the data. It is important for site

managers to know whose rent checks have been entered into the database and when, as well as up-to-date addresses and phone numbers for all residents. By allowing easier access to data, agencies can increase the effectiveness of their site managers.

However, there are additional issues that arise when giving staff access to the data. Beam discusses this in "Purchasing Computer Systems: An Executive's Guide" in the *Journal of Housing*. Staff need to be willing to take time from their jobs to learn the new system and help transition the data. If staff members do not want to use the new system they may "fail to learn how it works", or claim that it does not do what they need it to do, or they may simply not use it. People are often averse to change, and installing a new computer system, especially where there was previously none at all, can be a significant change. One of the best ways to engender support is to involve others in the decision making process, or at least keep them informed (Beam, 1990). Once you have support for the system, there are other issues that involve the staff.

McDonald discussed these issues in the September/October 1993 issue of the *Journal of Housing*. He pointed out that these issues include the possibility that some of the staff don't know how to type. Some of the staff may have never been exposed to computers, or the use of computers in their work. Many will believe that the computer is not necessary to their work and will only make things more difficult. Others just do not like computers and will not want to go near them.

Other difficulties that he points out include,

1. While attending training sessions to learn the new system, employees may fall behind in their jobs.
2. Employees may have to skip training sessions in order to do their jobs, thereby not really learning how the system works. This is especially problematic when the agency pays a lot of money for the training sessions.
3. Training often occurs too long before the system is implemented, so that by the time the new system is operational, staff need refresher courses.
4. There is often a need to enter data that are not in digital form — in addition to normal jobs. (McDonald, 1993)

These are all things that must be considered from the very beginning. Doing so will make employees more amenable to the idea of the new system, and more interested in attending training sessions. If employees are not included until later in the process, they may feel disenfranchised. They might be concerned that they will lose their jobs, or some of their power. By including them at the beginning you reinforce the attitude that they are important and that they are helping to shape this new system. These issues should be addressed during the initial Request For Proposals (RFP), so that there are funds available to hire additional help during the transition period, if necessary (McDonald, 1993).

During the transition period difficulties with moving from an old system to a new one are often encountered. For instance, the new system may not be performing the required or stated tasks, or there may be extreme difficulties in converting old data to the new format. Such problems might occur if there is not a data dictionary included with the old system. Without a dictionary, the agency may not know which columns in which files actually represent the data in the fields on the screen. In other cases, the agency may have signed an earlier contract that prevents the new company from reviewing the old system. This would prevent them from examining the current data structure to determine which data can be electronically transferred to the new system. Additionally, technical details such as length or type of data fields might get in the way of a successful transition (McDonald, 1993;Beam, 1990).

McDonald also discusses the need for goals and plans before beginning to look for a computer system. This is very similar to Rockart's discussion of reengineering the information systems department in a large company. (Rockart lectures, 1995)

Issues in planning where IS would be helpful

Joseph Ferriera Jr. discusses database management needs of planners in "Database Management Tools for Planning" in the APA Journal. He begins to address the issues of the complexities that arise when data cannot be stored adequately as a single table, when the data are constantly changing, when data must be shared by multiple users or agencies and the difficulty involved in transforming the data into other formats and combining it with data from other sources. He also examines the complexities involved when the data must be accessed interactively. Ferreira discusses these issues in general planning terms, however these are all issues that arise for the planners in housing agencies as well. Ferreira focuses on tools to meet planners' needs.

The software package market for housing agencies is largely based around the forms generation packages that Ferreira first discusses. Large database managers almost always provide the capability of on-screen forms for data entry and retrieval. The market that is examined in this thesis is in part a niche market of prewritten forms packages that are specialized for the housing industry. Some of the packages I examine are based on forms generation packages that serve as front ends for databases such as Dbase IV and Oracle. Ferriera discusses the use of these forms that "use visual cues to help improve data entry speed and reduce training time. In some cases, they also provide extensive 'edit checking' capabilities to reduce the likelihood of entering inaccurate data." (Ferreira, 1990)

Ferreira also discusses the comparison of various database tools. He points out that "one should not view each package simply as providing more or fewer features. Rather, one should think of these tools as incorporating different approaches to the same set of tradeoffs." (Ferreira, 1990)

Ferreira also points out that the planner should understand the format of his or her data and how the data are stored. This allows the planner to take advantage of relations in the data that may not be readily apparent. Understanding and knowledge of various tools and the format of the data, allows planners to efficiently extract information from their data (Ferreira, 1990).

While Ferreira discusses planning tools and issues, several other authors discuss new issues and programs in the housing field that make data automation invaluable. In addition to the programs discussed below, it is useful to remember that the impending decentralization of the Federal Government is going to have a large effect on housing

agencies. While the number and length of reports to the Federal government will probably decrease, the local and state agencies will have much greater autonomy and will have greater responsibility for managing their own programs.

Some of the recent policies and programs that increase the amount of reporting that agencies must do, and that introduce procedural change into housing agencies, are listed below. These new programs make a good information system invaluable. According to McDonald, the following changes require automation to keep up the needed information:

Project Based Budgeting and Accounting
Family Self Sufficiency
Public Housing Management Assessment Program (PHMAP)
HUD Form 50058
Comprehensive Grant

Although these cause an added workload, a well designed, integrated management information system, efficiently and effectively implemented, can make the increased work load manageable (McDonald, 1993).

Other programs that rely on the management capabilities of a housing agency include the comprehensive improvement programs. These programs implemented in the last 5 years are generally more complex and diverse than other programs agencies have previously been responsible for. These programs often include social service issues that have different informational needs than previous programs undertaken by housing authorities (Burgess, 1993).

Some Public Housing Authority (PHA) software can be very helpful in managing these programs. However, I do not discuss the management aspect of the software. Rather, this thesis focuses more on the information that such software can produce. I especially focus on the information that is useful to the planning and implementation of new comprehensive programs. The Boston Housing Authority's URD project is an example of one of these programs.

Burgess and Alexander also point out that there are many ethical privacy issues that arise as a result of increased information technology. These issues are very complex, and opinions about how best to handle them depend upon one's world views. Each person's ethics and morals are different and therefore each person views the ethical issues

surrounding information technology differently (Burgess, 1993.; Alexander, 1990). While this is a very interesting issue, it is too large to be addressed in this paper.

Chapter Three Databases and Software

The Evolution of Databases

First I will define the difference between a database and a database system. When referring to a database we mean to indicate the actual data, in files. The size of such a database is dependent on how much data the system contains and how the data are stored. A database system is the hardware and software that hold the data and allow it to be accessed.

Database systems have been constantly evolving over the years. At first, the management of data was not done separately from the computer program that used the data. A database was a program that managed data. Eventually during the 60's and 70's separate tools were developed to manipulate the data. These tools were then brought together in libraries and eventually became the software used today. During the late 70's and 80's, several mid-range computing systems were designed with the database software and access mechanisms as a part of the operating system. This included such machines as IBM's AS/400 and the System 36. However, as general purpose micro and mini computers have proliferated, it has become more common for the database programs to be independent of the hardware. In the 1980's, relational databases began to become practical. Such databases were theorized many years before the technology was available to actually build them. These have been developed into full scale relational databases that run on many different platforms. Popular industrial strength systems are Informix, Sybase and Oracle.

The reduced cost of storage and computing plus the increased amount of networking has made database systems and distributed access to data more and more economical, and a vast array of tools have been developed as a result. This thesis will focus on relational databases, and all queries are done on an Oracle server using a command line interface.

Networking

Increased networking and client/server capabilities are more reasons why automating housing authorities is more feasible and desirable than ever before. Networking is the means by which computers are able to communicate with each other and obtain information from each other, often in manners transparent to the end user. Networking allows users to work with each other electronically. Computers are no longer completely stand-alone isolated tools, but rather aid communication and interaction between users.

With a networked group of computers, users may share documents and information instantaneously and with few intermediaries. A planner need not print out a paper or report and then have it delivered to someone else in order to review it, and then she need not type in the corrections herself. With today's technology the planner need only indicate to someone where the report is located and that person can directly add annotations and suggestions at her convenience.

The client/server model adds to this convenience of shared data and multiple points of access. In the past each person had her own computer which held her data, programs and files. In order to access her data, she would need to be working on that specific computer. Networking computers together eases this restriction of access, as people are able to access data on other computers remotely. The client/server model takes this one step further. In this model all of a person's files and programs are stored on a remote machine, called the server. When a user wants to access their data they do so from any remote client that communicates with the server, and thereby have instant access to their files. The programs are also on a remote server, but they are run on the local client. In this way, compute-intensive programs do not slow down the machine where the data are kept. This also allows for the flexibility of running a compute-intensive program on a separate machine, and only display the output on the local machine. This adds to the flexibility of being able to run several programs at once.

The flexibility of these client/server systems is exploited in database management when the data are stored on a remote server while the user is able to run local programs that use standard, efficient protocols to query and manipulate the data. This allows users to use individual "front-ends" tuned to each users needs and computing equipment. Because the data are stored remotely, various people within the housing agency are able to use different programs to access the same programs.

This client/server model also increases the mobility of the computer user. The planner is able to access the data she wants, or her personal files, from wherever she is. There might be a terminal in her office, one in the conference room, and terminals at each housing site. Other planners will have terminals in their offices as well. In this way the planner is not dependent on a single machine or location. With some systems she may even be able to access her files from home.

SQL and Relational Databases

Date explains that a database system is "essentially nothing more than a computerized record-keeping system;" a relational system is one where the data are perceived by the end user as tables, and the operators available to the typical end user allow the user to generate new tables from old ones (Date, 1986). Stated simply, the user is able to relate tables to each other and to select subsets, groupings and unions of the existing tables. Many of the packages discussed herein provide these relational capabilities. There have been many languages developed to access the data in relational databases. Over the years the SQL language has developed as the standard. This is a language for transforming data into various formats that suit ever-changing needs and hypotheses. This tool opens up a much richer set of possibilities than is possible through flat file spreadsheets (Ferreira, 1990). In the past, it was necessary to pre-define queries that were going to be asked and organize the data so the queries could best be answered. SQL brings us away from that necessity and allows ad hoc queries of the data. The following is a basic introduction to SQL.

Here is a portion of a table of housing units that a housing agency might own:

Units:

UINDEX	HOUSENO	STRENAME	APARNO	HHINDEX
MM32490	4	Ames St	6	AH23433
MM23494	6	Smith Way	3	AH90343
MM23454	4	Ames St	2	AH92384
MM42393	3423	Huntington	32	AH89734

One might want to view only the people who live at a particular address. To do this one would select a subset of the rows. This is done by asking the database for the unit, address, apartment and tenant number for only those rows where the address equals "4 Ames St" for example. The actual SQL query that you would use looks like this:

```
SELECT UINDEX, HOUSENO, STRENAME, APARNO, HHINDEX
FROM UNITS
WHERE STRENAME = "Ames St" AND HOUSENO = "4";
```

And from our table, the result would be:

Result:

UINDEX	HOUSENO	STRENAME	APARNO	HHINDEX
MM32490	4	Ames St	6	AH23433
MM23454	4	Ames St	2	AH92384

This listing could then be used to contact everyone in a specific building about a specific local event, like the exterminator coming. Or a similar request could be made for everyone on a particular street, to inform them of an upcoming street party.

One can imagine that any spreadsheet or database tool can pull out the specified rows. However, we might want to combine this information with information from other tables for example, a table describing the name and other tenants information. We would not want to put all of this information into the units file because the tenants in each unit change. Storing only the tenant ID in the units file and the rest of the data elsewhere, helps maintain data integrity. In this way, the units table does not need to be accessed when personal characteristics about the tenants change, only the tenants table need be. Such a table of tenants might look like this:

Tenants:

HHINDEX	LNAME	FNAME	AGE	RACE
AH23433	Smith	Jerry	23	Black
AH92384	Jones	Angela	34	White
AH02349	Tanner	Richard	45	Hispanic
AH90234	Cirrilo	Lore	33	White

The table about housing units does not tell us anything about the tenants themselves. Since our information is now stored in several tables, we have to bring them together in order to do many tasks, such as creating a mailing list with tenant names. When the data from two or more tables are brought together it is called a "join". For example, if we were to send something to the tenants we would need the name of the tenant in addition to the address, but we wouldn't need the unit number or the household number. To get the information we need, we would ask for the head of household's first and last names, and the address and apartment number. However, we need to tell the database which names go with which apartment. To do this, the tables must have a field in common to match on. In this case the field is "HHINDEX". Therefore, we tell the database to match the HHINDEX columns together. Then, it will give us the address for the row with the corresponding HHINDEX. Such a query and the results would look like the following:

```
SELECT FNAME, LNAME, HOUSENO, STRENAME, APARNO
FROM UNITS U, TENANTS T
WHERE U.HHINDEX = T.HHINDEX;
```

FNAME	LNAME	HOUSENO	STRENAME	APARNO
Jerry	Smith	4	Ames St	6
Angela	Jones	4	Ames St	2

This query will give us all the rows where there is an address and name that match. This sort of join illustrates the versatility of SQL. We do not need to know what order the data are stored in, or whether it would be faster to go down the list of tenants and find a corresponding address for each tenant, or whether we should find the corresponding tenant name for each address. We tell the database what information we want and it decides how to retrieve the information. We also do not need to know in advance what information we are going to need or what queries we are going to ask. We can decide "on the fly." With older database systems it was necessary to predefine all the queries that were going to be asked of it. These queries would then be written into the software and the user would select them from a menu. Some questions were also difficult, time consuming or even impossible to answer because of the way the data were stored. SQL is powerful enough that the user does not need to be aware of such fundamentals.

The user does, however, need to have some understanding of the data, in order to know which tables contain what information and how the tables may be related to each other. This is the type of information that can be obtained from a data dictionary. Data dictionaries are discussed later in this section.

There are several other types of databases for which one uses different languages and commands, some of which include flat file, inverted list, hierarchic and network databases. The BHA system is a hierarchic one and there were several conversion steps that needed to be taken to bring the data from that system to a relational database. I will not go into these types of databases in any depth here; However, it should be noted that the software packages I look at in this thesis include several of these different database types.

The key operations that all database systems must be able to perform include adding new files to the database, inserting new data into existing files, retrieving data from existing files, updating data in files, deleting data from files and removing files from the database (Date, 1986). It is possible to control access to the database allowing different end users to

perform different tasks. One end user may be able only to view the data in various groupings, while another has the ability to add, delete or update records while reserving the power to add and delete whole files for someone such as a system administrator. This is how many of the software packages for housing agencies work.

In addition to tables, each database should have a data dictionary. A data dictionary is a plain file which describes what is in the database. It generally lists what files are in the database and describes the information they hold, as well as explaining what data are in each column, in what format and a description about how each field is used.

A data dictionary for a database holding the above tables might look something like this:

UNITS: holds information about each individual unit in the possession of the BHA.
UINDEX - char(6), the unique identifier for each unit
HOUSENO - char(5), the street address number of the building the unit is located in.
STRENAME - char (20), the name of the street the building is located on.
APARNO - char(8), the apartment number of the unit
HHINDEX - Char(6), the unique identifier for the current household, blank if the unit is empty

TENANTS: holds information about each head-of-household that lives in BHA housing.
HHINDEX - char(6), the unique identifier for the household
LNAME - char(10), last name of the head of household
FNAME - char(10), first name of the head of household
AGE - int(3), age in years of the head of household
RACE - char(10), race of the head of household

Choosing DBMS Software

Before looking at the software programs written for housing agencies, it is useful to examine some of the issues involved in choosing a computer system. Several authors discuss the benefits of systems that are not dependent on a specific computer platform, and the general flexibility of programs.

When choosing a new computer system, the type of system and its capabilities are very important. Edward McDonald recommends an "open" or "4th generation language" system. These systems allow easy transfer between platforms, are based on relational databases and utilize SQL. He defines a Fourth Generation hardware platform as one that has a UNIX based operating system, and a software package as 4GL, if it is "portable across many hardware platforms without major changes in functionality and uses embedded Standard

Query Language." (McDonald 1993)[2] McDonald recommends these types of packages because it allows for easy expansion and change of the system.

Having SQL capabilities also allows the user to ask more complex and interesting questions of the data then something like Dbase or a spreadsheet does.

Jorgensen, Manis & Schaffer discuss several issues of relational database management. Some of their key points include:

1. Data should be kept in flat ASCII tables (files), not binary, so it can be viewed outside the program, and a special program is not depended upon to decode the data.
2. Programming should be done in a 4th generation programming language, the most powerful of which is the UNIX shell.
3. Programs should be small and should pass data on to other programs.
4. All programs should be integrated with a common interface with both users and data. All of the tools should look more or less the same for quick learning and easy use.
5. Software systems should be built to meet interface standards so that software can be shared.

Networks should be used to tie together computers and software so that simple tables of data can be passed from one to another (Manis, Schaffer & Jorgensen, 1988) .

Although these are certainly not all the issues that should be considered when reviewing computer tools for planning they are a good place to start. There are several other issues raised in the following discussion of software written specifically for housing authorities.

Software for Housing Authorities

Today computers can be used for many purposes including financial tracking and management, management of current properties and planning of future projects. Computer resources have been found to be a great source of assistance when preparing elaborate proposals for federal and state funding. Until ten years or so ago, computers were too expensive to be used by housing agencies and other nonprofits for any purpose other than for financial tracking. In many agencies, such as the Boston Housing Authority, IS was therefore placed under the finance department. IS is now breaking out of this trend with

[2] Today, some PC-compatible systems are adequate in addition to UNIX platforms.

today's more advanced computers. Software that allows total agency management and is developed specifically for housing authorities is a growing field.

The best source that I have found for this software is the "Computer Services Review", published every year in the *Journal of Housing*. This article lists many different products, including turnkey computer database systems, handheld inspection computers, and construction and bid management software. This portion of my thesis will focus mainly on software used to manage housing authorities and their tenant lists. I obtained all names and phone numbers from the 1993, 1994 and 1995 *JOH* "Computer Services Review." They note at the beginning of the reviews that they do not recommend any individual product and that they asked "vendors who specialize in computer products and services for the housing and community development field to supply descriptions of their products" (Rajah, 1995)

Many issues are important in deciding which software is best for a given agency. Some of these issues are based on the software itself, some involve pricing; other issues include potential inter-operability with software at other agencies, future applications and stability of the software provider.

When deciding which piece of software to purchase it is important to focus on several items. One of these is recognizing information requirements: What are the HUD requirements now and what are they likely to be in a few years? What information do the directors of the agency need for management and future planning? What information is needed by case workers and what information by site managers? Edward McDonald suggests that one should also take into account the availability of software support in the future and anticipated costs for the next five years as well as expandability of the system.

HUD requirements are important to all agencies that receive funding from the Federal government. Such agencies must issue reports to HUD on a regular basis with information about how they are spending the moneys they are given and on whom they are being spent. Special attention must be paid to Section 8 recipients, so those reports are fairly detailed. Most software providers who specialize in software for housing agencies are aware of these constraints and guarantee that their software produces reports suitable for HUD reporting; many offer software updates at regular intervals that keep the software in compliance with HUD standards.

A variety of software packages are available to housing agencies but, choosing the appropriate packaged based on an agencies particular needs is very important. In order to give the reader a feel for the range of software available today, I will distinguish between the software products on the basis of phone interviews and promotion packets.

During my research I focused only on software packages that were designed for use by housing authorities. I looked at the databases, the platforms the software ran on and what sort of support services the companies provided. Companies had significant differences in attitude towards discussing their products with me. Some were very secretive and proprietary, while others were happy to discuss their software. I called 24 different companies. Two of the companies do not have the type of software I was interested in, but make software for managing rehabilitation and maintenance aspects of the housing industry only. Four places neither called me back nor responded to follow-up calls, and I was never able to reach one. This left seventeen companies in my study group. In addition to the phone conversations with these 17, I received seven packets of written information from my requests and 3 demo disks. These 17 companies are listed in the table at the end of this section.

My analysis and conclusions are based on these phone interviews, packets, demo disks, experts in the aforementioned "Computer Services Review" and information from various advertisements. The only software from my study group that I have worked with personally is Memory Lane Systems' software at the Metropolitan Boston Housing Partnership (MBHP).

My survey questions included the name of the software, what hardware it runs on, and if it is multi-user, modular, turnkey, menu driven, and to what extent it interacts with 3rd party software. Also, for comparison purposes, I asked the following questions: What language is the package written in? Was it completely developed in house, or is it based on a commercial package such as Dbase or FoxPro? Who makes customizations to the software, if any, and what changes or updates does the company regularly perform? Is there support for Geographic Information Systems (GIS, mapping capabilities), or SQL? What support is offered for transitioning data from the old system and what size customers do you generally target?

The information I got and its accuracy may have varied from company to company depending on whom I spoke to. Some representatives seemed very familiar with all aspects

of their software and computers in general, while others seemed more like salespeople who only knew about their own systems. For example, when I asked about SQL, if the representative indicated that they had no knowledge of SQL, then I assumed that their software did not support SQL programming and that they were not a developer or programmer.

A summary of the results of my queries are found in Table 3-1. This table lists the software companies that I reviewed in alphabetical order, along with the name of their software. It then lists the responses that I received to a selection of the questions for each company. Some of the categories listed include what hardware the software runs on, if the software is menu based, what language it is written in, who makes changes to the software, whether it has SQL capabilities as well as what support they provide for transfer as well as many other categories.

When I asked what platforms their software ran on, I received a variety of answers including "all platforms", PCs, networked PCs, the AS/400, and various UNIX platforms. As a summary, six companies responded that their software worked only on PCs or networks of PCs. Three responded that they ran only on UNIX boxes. Five agencies mentioned that they ran on AS/400s and two mentioned that they ran on mainframes or minicomputers in addition to more current platforms. Five agencies said that their software ran on both UNIX boxes and PCs. Not one company mentioned anything made by Apple. I assume that when a company said PCs they used the common term meaning IBM compatible machines, and did not mean just any personal computer since Macintoshes have very different operating systems than other machines. Most companies indicated that they could provide "turnkey solutions" that is, they would also assist in the selection and purchase of the necessary hardware.

Most of the representatives said that they developed their own software that was not based on anyone else's. However, some of these people also said that their software is written in "clipper" which would indicate that it is based on Dbase. Four of the software packages are based on Dbase, and one is based on FoxPro. The other packages are written in RPG, C, C++, COBOL, Basic, or "4GL languages."

All the packages can be considered "modular". That is, there are different packages that you can buy separately that work together. You can buy either some or all of the modules. This makes it easier for a smaller company to buy part of the system now, and

then another part in a year, and keep adding to it, as more money becomes available to invest in the computer system. However, this process also locks the agency into working with one vender, because the different vendors' modules do not work with each other.

Companies range from offering 3 modules: Accounting, Tenant Management and Office Management, or four modules: Tenant, Property, Financial and Office Management, to 10 to 15 modules. The ten to fifteen module packages generally include, but are not limited to: General Ledger, Purchasing, Inventory, Fixed Assets, Work Orders, Payroll/Personnel, Accounts Payable, Waiting list, Section 8, Tenants, Utility Tracking, Unit/Property Management. All of the vendors claim that their packages are fully integrated. This generally means that the daily user cannot tell where one package ends and the next one begins. It also generally means that they access the same data. That is, if you enter the data once, you do not need to enter it again in a different part of the software. Several vendors mentioned that access to the various modules can be limited by employee, thereby giving employees access only to the data that they need for their job. This protects the tenant's privacy and helps insure data integrity. For example, the maintenance worker can see his work orders and other related information, but not the family composition or income of the unit he is to repair. Similarly, the intake worker can see the waiting list and the clients information, but is unable to see the work order queue or payroll.

Most software packages integrate with third party software. Usually the third party software is a word processor. Nine of the vendors mentioned the use of Word Perfect in particular. Three additional vendors indicated that they would work with third party word processors, or "any word processor." Two vendors said that their software was a standalone product and did not interact with other software. Many vendors also indicated that their software would work with Lotus 123 and a suite of other commercial programs. One company's idea of an office management module is to allow the housing agency to pick from a suite of word processing, spreadsheet, relational databases managers and windows applications. The vendor then promises to turn the chosen applications into an integrated office management system.

My queries about mapping software, or Geographic Information Systems (GIS) led to answers that ranged from, "I'm not sure what you mean by that... Oh, no, we don't have that" to "We have a module for that under development." None of the vendors I queried had a working mapping module. However, a few were working on it, or would be willing

to develop one if the housing authority requested it. One vendor pointed out that it was very easy to download their data into plain ASCII text and it could thereby be loaded into a desktop GIS program fairly easily. I suspect that some of the other packages could do the same. As desktop mapping becomes a larger part of planning in public housing authorities, desktop mapping tools will become one of the applications available for integration into housing authority software.

I asked many of the vendors about how easy it is to download plain ASCII flat files of data from their database. How difficult they indicated this was tended to be related to what sort of database they were using. One company indicated that they were using an ISAM (index sequential access method) key structured database written in COBOL, and that a report would have to be written by the software company to get the information out. Theirs was the least flexible of the programs, their software had only predefined forms, and there was no programming language in which an in-house programmer could write additional forms. The agency would have to go back to the vendor for that. They indicated that they did have a report writer, but that it was not very good and they don't recommend it. It seemed that the more advanced the software, the easier it was to get data out in the way the customer wanted. The packages that are based on relational databases seem to be generally more flexible in producing reports the user wants in formats that are desired.

I tried to get a feel for each company, for their stability and reliability, however this impression is based generally on only one phone call; in most cases I spoke to only one employee. However, there was enough similarity of capabilities, context and strengths and weaknesses that I felt confident in understanding the basic features, target markets and different emphasis of each package.

The BHA uses software by Creative Computer Systems that is running on a PICK Unidata database. It is described further in a later section.

Company	Software Name	Hardware	Turnkey	Menu Base	Language	In House Developed	Who Makes Changes	Support for Changes?	Based on 3rd Party Software	Interaction With Other Software
Advanced Integrated Technology		AS/400, UNIX	Y	Y	RPG	Y	They do	HUD, fixes and enhancements	N	will consider
Alpha Data Systems	The Authority	UNIX	Y	Y	Their own	Y	They do	they will customize	N	will customize
Automation Research Systems Ltd.	Public Housing Integrated Management	UNIX, anything	Y	Y	C++	Y	They do	no customization	Oracle	yes
Creative Computer Solutions	PHA Software System	UNIX, Unidata database	N	Y	C?	Y	User or they do	yes	Unidata Database	yes
Hawkins, Ash, Baptie, Inc.	Housing Managemnet Solutions	UNIX, PCs	Y	Y	COBOL	Y	They do	yes and updates	Y	Yes
Lindsey Software Sys.	Lindsey Housing Software	PCs	Y	Y	clipper	Y	can customize fields	yes	No	yes
Management Computer Services	MCS Housing Software	UNIX, PCs	Y	Y	COBOL	Y	They do	yes, and enhancements	Yes	Yes
Management Research Corporation	HOMES	PCs	Y	Y	Clipper	Y		updates	Yes	Yes
Memory Lane Systems	Emphasis	any	?	Y	C	Y	can purchase source	yes	No	Eventually
Modern Software Technology	PHA Software	AS/400	N	Y	RPG	Y	customer can change	yes	db2 software	db2 software
RAM Enterprises	HUD Tailored Software	AS/400, Sys 36, PCs	Y	Y	RPG2	Y	can purchase source	only if not modified	No	Yes
Tenmast Corp	Tenmast Houing Management SW	PCs	Y	Y	Clipper	Y	they do, HUD changes	yes	dbase	WP, Lotus, Quattro Pro
Project Data Systems	Micro HUD, MicroRent, Facts	PCs	Y	Y	C	Y	they do		No	
Wright, Darnell and Rector	W.D. & R Software	PCs & Mainframes	Y	T	Basic	Y	they do	HUD changes	No	Word Perfct, Lotus
Tri Tech	Computing Housing Authority	PC, UNIX	Y	Y	COBOL, ISAM key	Y	they do	HUD, only if for all	ISAM struct.	No

Table 3-1 Summary of Software Packages

Company	Word Prcoecessor?	Mapping?	flat files	Dbase	SQL	Report Generator	Support for Transfer	General Support	Size of Authorities
Advanced Integrated Technology	3rd Party	in development	Yes	No	Yes	Y	Full, may cost extra		all sizes
Alpha Data Systems	Word Perfect	no, would consider		No	No				
Automation Research Systems Ltd.	Word Perfect or preference	demographics, easy to integrate	Yes	No	Yes	Y	Full	Full	all sizes
Creative Computer Solutions	doesn't come with	in development	?	?	Yes	Y	Yes	Full installation, telephone	mostly large authorities
Hawkins, Ash, Baptie, Inc.	Word Perfect, Word, Uniplax	No	Yes		Yes	Yes	they help, can do	Full training	medium sized authorities
Lindsey Software Sys.	yes	in development	No	Y	No	Yes	No		all sizes
Management Computer Services	Word Perfect & Lotus	No		Y	No	Yes	Not much		all sizers up to 128 users
Management Research Corporation	Word Perfect	No		Y		Y	Yes		
Memory Lane Systems	Yes	No	?		?	Yes	Yes		mostly mid-sized
Modern Software Technology	on AS/400	No	yes	N	Yes	Yes	Yes	some	large
RAM Enterprises	Word Perfect, spreadsheets	No	Yes	N	on AS400	Yes	Some		
Tenmast Corp	Word Perfect	No	Yes	Y	No	Yes	depending on system		all sizes
Project Data Systems		No		N	N	?	Some		
Wright, Darnell and Rector	Word Perfect	No		N	N	Y	Yes		
Tri Tech	No	No	No	N	N	N	Yes		generally small

Table 3-1 Summary of Software Packages

Chapter Four Planning Questions and Information Requirements

What people want to ask

Running a public housing authority is a very complicated and time-consuming process. There is often less money than one would like, and not enough staff to decide where to spend the money. The staff, and particularly planners on the staff, have many questions that they would like to have answered before they decide how to allocate the money. Due to the fact that the resources available for research and analysis in housing authorities are scarce, planning questions are often not answered, or a large burden to answer. With the proper training, and tools such as a relational database and SQL, these questions could be answered much more efficiently and effectively then they generally are today.

During our project with the BHA we worked with various planners and other employees to find out what information would aid them with their work on the Mission Main Urban Revitalization Demonstration project (URD), particularly information that they cannot currently access readily themselves. This, in addition to my other work and studies with housing authorities, gave me an idea of the types of information that are generally useful in housing authorities for the planning and research activities that may not yet be built into the administration of the housing authority.

The URD project was designed to incorporate all aspects of a housing authority and all aspects of people's lives. The Boston Housing Authority was empowered through this project to work with social service agencies and develop their own social programs as well as to work with the physical aspects of their housing. This program allowed greater flexibility and tradeoffs in funding in the hopes that the BHA could make the best decisions about where the money was most needed. To these ends the BHA became involved with aspects of their tenants lives that were previously barred to them. Because the BHA was given more freedom in deciding how to use their funding there was a need for more studies and analysis of the needs of their tenants. The questions mentioned in the following sections are all things that the BHA may want to ask regarding their URD project. However, they are also questions that applicable to housing agencies around the country.

Planners may wish to ask questions about the composition of their housing developments when deciding where to spend money. They may wish to know about

movement patterns within their developments, such as which buildings have the longest average tenancies and which have the shortest, or they may wish to know, where do the people live who have lived in the development the longest? Once they have the answers to these questions, the planner may wish to actually go look at the buildings and see if they can understand *why* some buildings have longer tenancies than others.

Planners may also wish to know if different types of populations live in certain areas within the development and where. They may wish to know where the children live in order to locate an afterschool program or daycare. Another question may be, where do the children who go to a particular school live? With this information they may be able to help the parents in forming support groups or local parent organizations so they have support in confronting the school authorities (BHA Interviews). The planner might want to know how many elderly people are in the development and where they live, so that she can design and site a senior citizens center.

If there are feelings of animosity between the neighborhood and the development the planner may wish to take a look at the demographics of the two areas to see if they might account for the tensions. Census data and the data collected by the authority would allow the planner to compare facts about race, income, and possibly even educational levels. With this sort of insight the planner may know better how to ease the tensions between the housing development and the surrounding neighborhoods. She would know what sort of image would be best to create, or what sort of joint programs might work best.

Parts of these studies are quite simple, while others are complex. Obtaining a count of people with one characteristic or another is fairly straightforward in just about any system the agency might use. The complexity comes in when one starts to combine characteristics, or wishes to relate population characteristics with spatial location. These questions can also become iterative. For example, if the planner wants to compare the characteristics of the development with the surrounding areas, she would first have to come up with a hypothesis about where the differences lie, and then test it out. She might first say, "I think there are tensions between the two groups because of racial differences." However, she might test this out and find that the racial distributions of the two groups are more similar than she thought. In that case she might suggest, "It is the income differential that is the problem." Upon finding some correlation there, but not a large one, she might realize that there are specific places where people are employed, and do queries and research about

that. It was only through an iterative process that she came to this realization however. Because the planner had instant access to these data, she was able to ask, and immediately modify her questions as conclusions became apparent.

I break planners' questions into five types of queries. These range from automation type queries that are commonly and easily answered today, to very complex questions that necessitate hiring a consultant to do a study. Although I will briefly mention each type, I will not focus on either of these extremes. Since neither of these types require a special tool or training on the part of a planner, they are beyond the scope of this thesis.

The first category is the automation type of question, such as "How many people are on the waiting list?" or "What are the addresses of our tenants?" These sorts of questions are the type that need to be answered constantly at the housing authority, and are answered on a day to day basis. Authorities must produce reports indicating the lengths of their waiting lists at regular intervals, and they often need to have tenant addresses on hand. These types of questions are very simple to answer and are very straightforward in any system the agency might have. Therefore they will not be addressed here, except as basic examples of how to use the SQL query language.

The second type of question includes those that are easy to answer if you have the data available to you. This might be questions such as "How many of our tenants are black?" or a slightly more complex question such as "How many of our tenants are of which race?" or "How many children live in units that we subsidize?" This might also include queries like "What is the average rent our tenants pay?" or "List the tenants that are late with their rent this month." These are questions that are commonly asked in housing agencies today, but often take too much trouble for people to find answers on a regular basis. It is usually possible for an authority to answer them today, but the process is not always as simple as one might prefer. With most of today's systems, an administrator can not be sitting at her desk, think of a question, and find the answer. She must submit a request to the MIS department asking to have the question answered for her. Often people prefer not to submit requests for such a small amount of information, or they are not able to receive the answers in a timely manner. In these cases they will often use their personal knowledge to estimate an answer. If these administrators and planners had desktop access to these data and some understanding of how the data are structured, they might be able to find the answers for

themselves more easily. Some of these questions can be answered by a forms-like interface to the database as well, thereby minimizing the need for training of staff.

The third type of question is the type where a planner might be looking for variations on a theme. None of the questions in and of themselves are very complex, but the process involves asking a series of questions, the answers to which spur further questions. These questions are difficult to answer today more because of the information structure and the processes for getting questions answered, than because of the actual complexity of the data itself.

For instance, a planner might want to look at the information regarding tenant employment. First he will want to see what categories of employment are being used. The results of this query might show that there are many categories of income that need to be standardized. In that case the planner would group the categories so there are just a few types of income to look at, perhaps gainful employment, insurance, welfare and no income. Then he will see how tenants fall into these categories. These investigations might require looking at the data and grouping it in different ways to see what makes the most sense. The planner then might decide to do groupings by race, age, location, single-headed households, or whatever seems to make sense with the data he has. The difficulty with this process is that it is necessary to do queries, examine the results and then do more queries, perhaps on some subset of the original data identified in an earlier step. With the typical MIS department today, this involves requesting reports, waiting for the results and then requesting more reports. The bureaucracy involved turns this simple desktop study into a complex process often lasting weeks.

With a desktop query system and access to the database, these sorts of queries could be done by the planners themselves. They could query the database immediately upon thinking of the question, receive answers in real time (minutes instead of weeks) and then be spurred to ask additional questions. This sort of real-time research is more conducive to innovative ideas. Because the planner is not asking someone else to find the answers for her, she is likely to ask more questions and more open-ended questions, hoping to come up with something new. She does not have to worry about wasting anyone's time but her own.

A query such as "What percentage of current tenants are gainfully employed?" can be fairly complex depending on the format and reliability of the data. For example, one might

begin by categorizing income sources. This involves understanding how income sources are currently categorized and whether they are standardized. These must be then grouped into a few large, rational categories. This involves knowledge in the field and the use of personal judgment. Then one must examine who is in the database. It is likely that not every entry in the database represents a current tenant. A determination of how to select current tenants must be made. In the case of the BHA data, the information about employment is in a separate table from the information regarding which tenants are current. The planner needs to have an understanding of how these two tables relate to each other in order to bring the data together.

This is a fairly complex process, but if the person doing the queries is familiar with both the software and the data, it should not be very time-consuming. This sort of query involves professional judgment, in deciding what category the income source would fall under, as well as knowledge about the data in order to understand that not everyone listed is a current tenant, and how to determine which are the current tenants.

Having done this query other questions might come to mind as well. The planner might then want to ask questions such as "How many of the current households have more than one person gainfully employed?" or "What are the percentages for the other categories?" The planner might be interested in time questions such as "How does the percentage of current employed tenants compare to the percentage of employed tenants five or ten years ago?" With the knowledge of the tool and the data that the user now has, she could then go on to answer these and other queries.

The fourth type of question is more complex. This is the sort of question that requires data from separate tables to be brought together and a special analytic tool, like access to a relational database or mapping tool. These queries might include: "How many tenants in building 1 are black? And are the tenants in our buildings segregated by race?" or "Do families who remain in the housing development for a long time prefer particular buildings and do they move between apartments a lot?" or questions of tenure.

Performing these analyses requires sufficient knowledge of the tool, and understanding of the data structure to enable the various datasets and tools to be integrated and combined effectively. The planner needs to know what data are found in which tables, how these tables are related to each other, and what are the limitations of the data. She also

must be familiar with her interface to the data, and its strengths and weaknesses. The planner must have used this tool before and understand its capabilities.

The planner should also be aware of the fact that the master files are often large, complex files of administrative data and the planner often needs to only view a subset of these data. They need to be aware of the added complexity due to the fact that the base data are constantly changing and being updated over time.

The fifth category is often more complex in that it involves bringing in data from outside sources. One might want to compare the demographics of a housing development with those of the surrounding neighborhood. Or they may wish to do a study of nearby social service agencies. When answering the question involves bringing in data from outside sources, a whole new series of issues of data compatibility and accessibility are introduced. These are beyond the scope of this thesis. I also include in this category questions that necessitate the hiring of a consultant to do extensive research and statistical studies that are beyond the ability of the average planner. Without the technologies available today, all but the simplest of the queries discussed in this section used to involve hiring a consultant in order to answer them.

Difficulties with Direct Access

I will now address some of the difficulties involved in answering the questions that planners would like to ask, and suggest some solutions. The agency must decide who is to have access to the data and what planning knowledge they need. Two options are to either train the computer staff to be more aware of planning needs, or to give the planners more data management skills. These issues are similar to the dilemmas faced by Chief Information Officers (CIO's) in large companies considering MIS decentralization. They often wonder if they should decentralize MIS, or whether the MIS staff should understand more about the business. Neither is really a bad idea, and both should be attempted, but the company or agency can only implement one solution to a significant extent. Hodgkinson discusses this in his unpublished article "The Role of the Corporate IT Function in the Federal IT Organization" (Unpublished). He lays out the merits of decentralizing the information technology staff of the corporation throughout the various departments and divisions, instead of keeping them in one central location. This encourages the IT staff to learn about the business and why they produce the information that they do. He has found that in many companies this solution is preferable. An analogous version of this in a housing agency would be to train the planning staff to use the computing tools to do queries themselves, rather than keep all the functionality of MIS in one central department.

If the decision is to keep the MIS staff at a centralized location, they must be educated about the issues found in the housing agency. If the MIS staff is to be more than just an MIS staff, they would need to have the knowledge to formulate and anticipate the types of questions that planners would want to have answered. If they were given the funding and staff support, a centralized access to data would be possible.

In this case, the MIS staff would not only maintain the information infrastructure of the agency, but would also anticipate questions and generate reports of their own on the operations of the agency. There are many drawbacks to a system such as this however. To begin with, the transformation would involve significant retraining of the current MIS staff or almost completely replacing the current MIS staff with planners who also know a great deal about computers. However, it is not easy to specialize in both of these fields and rarely would qualified candidates be the best in both areas. With this option, excellence is traded for flexibility. The MIS department would be separated from the rest of the agency and feel no particular loyalty to one department or another. Therefore their queries would not

necessarily be as intuitive as someone who works in a particular area, and wants information about that department.

The other option would be to train the staff throughout the agency to use the tools necessary to answer their questions and allow decentralized access to these data. This might be intimidating to the MIS department, as they might feel that their jobs are being taken from them, but a good manager could handle this. It takes both training and practice to learn to use complex relational databases, but this thesis has outlined their usefulness and many reasons why a planner should be proficient with these skills. By distributing these skills throughout the agency each person can focus on her own area of expertise and ask questions that are relevant to her work. In this way, the use of the information system would be both distributed and focused. Planning schools are beginning to train planners to have these skills today. Such courses are currently available to Masters students. Examples of such courses are MIT's Introduction to Computers in Public Management, A Workshop on Geographic Information Systems and Computer Based Analysis for Public Management. These classes are a good start, and a basis for learning the tools needed in planning, but it is only through practice that one can become proficient at using databases such as FoxPro or Access, or fully relational databases such as Oracle or Informix.

Planners are also the best people to be asking these questions because they are familiar with the many issues involved in formulating useful questions. However, there are many issues involved in answering complex planning queries. These issues include what data the agency keeps, how the data are stored, and what tools are available for accessing these data. Often the data exist, but belong to another agency — this adds another layer of complexity to the problem. There are many formats in which data can be stored causing translations from another agency's format to be quite complex. There are also often problems of data incompatibility, such as when agencies organize their data differently. If the data are from previous years, they might be stored off-line or in a complicated, hierarchical manner that makes it difficult to retrieve.

The many advances in database technology have greatly increased the feasibility of ad hoc queries on large databases. The ability to network and use a client/server interface where the client is itself a complete computer and not just a dumb terminal allows greater access to the data. New Graphical User Interfaces, or GUIs, makes these computer more user-friendly and intuitive to learn than older command-line interfaces. All of these, along

with a standard SQL language to use with the database, give researchers the ability to do real time iterative queries. They are able to formulate a question, receive a result from the database and based on the result, formulate a new, related question. In the past it was necessary for a researcher to submit a query and then wait, often several weeks, for the result to be returned to them. Because of the long delay between when the questions were submitted and when the answers received, the old process stifled thought flow and spontaneity. The process that can be achieved when researchers have immediate access to the database, is much smoother, and allows for a flowing thought process.

The tools for querying these data are also becoming simpler and easier to use. Where once it was necessary to write programs in order to extract desired information from the data, there are now tools that can assist in organizing the data in meaningful ways. The basic structure of relational databases, or Relational Database Management Systems (RDBMS), as opposed to older hierarchical or flatfile databases is one of these tools. Other tools include the programs that run as front ends to these databases. Front ends are programs that act as the user interface to the actual database. There can be hundreds of different front ends available for the same database. There are also programs that exist separately from the databases but behave much like a database, for example FoxPro or Access. These programs allow a user to manipulate the data much as they would with a relational database but they provide forms and menus to simplify the process for the user. The basics of these tools are not difficult to learn to use, and with a little practice a planner can become quite adept at using them.

The most common interfaces and programs are command line interfaces using SQL and forms browsers. A command line interface is the most basic and most powerful way to communicate with an RDBMS. Most databases today are designed to work with SQL, Structured Query Language. This is a very powerful, yet simple language that I described earlier, that allows users to query the database in a straightforward way. The user uses commands such as "SELECT columname FROM tablename ORDER BY columnname" Because the commands are logical English words, the language is relatively simple to learn and use. However, full mastery of the language takes a great deal of study and practice.

The forms browser is also relatively simple to learn the basics of. These packages serve as a front end to a relational database or stand alone as a semi-relational database, manipulating separate files. They allow the user to build the query by choosing options

from menus. The user is able to select the tables and columns and place conditions on them all through pointing and clicking. The program then builds an SQL query based on the choices the user made, and submits them to the database. These programs are less powerful than using a command line interface, but they are generally sufficient for most needs.

Older programs only had pre-programmed queries built in. These were acceptable for general use, but they essentially locked the planner into a certain set of questions. These programs allowed one to get hardcopy reports as well, but only of the preprogrammed queries. Current programs allow users to define their own queries and get hardcopies of the reports. They also make it possible to enter queries where an entire table is the output, which can then be exported to a spreadsheet program. Once there, the planner can manipulate the information in various ways to answer minor variations of his question.

The new interfaces provide nearly instantaneous results to simple queries, while more complex queries take slightly longer. Since the results are received in real time, researchers are able to ask several iterative queries in a short space of time without any break to the thought process. Because the databases produce results in real time and the interfaces are not difficult to learn to use, it is feasible for people other than researchers to use them to query data. It is now feasible for a planner in a housing authority to sit at her desk and query a remote database in real time.

Obstacles to Direct Access

There are, however, a few obstacles to overcome before planners can have direct access to the data.

One issue is data security. Most people feel that it is not always a good idea to allow unrestricted access to certain types of data. It is both logical and possible to restrict write access to data with most programs. However, it is often necessary to create temporary tables to achieve the necessary relations. Creating these tables requires write access to the database. One practice that allows people to have write access, yet not be able to alter the data, is to allow access only to a copy of the database. In this way, if there are any accidental deletions or changes made, the data can be restored from the original. It is also often necessary to allow people to see only a part of the data. For instance social service employees may be allowed to view data on a family's composition and address and phone number, but not be allowed to see race or income. On the other hand you might want

planners to be able to see all available data except Social Security Numbers, so that they are better able to make informed decisions regarding the future of the housing. This can also be handled in two ways. Many software packages allow you to set permissions so that only certain people, or groups of people can see certain information. This can also be dealt with by only putting some information in the copy of the database that people are allowed to use. For example you might translate Social Security Numbers into some other sort of unique identifier before allowing people access to the data.

This solution of allowing people access to a copy of the data solves other difficulties as well. Often large housing authorities keep their data on a mainframe or minicomputer. These computers are accessible only at one point directly, and additionally through remote dialup connections. Usually only a few people have access to this machine and data and anyone who has queries for the data must submit them to these few people. As a result it often takes up to several weeks to get the results back. These machines also work on a time-sharing principle such that only one program may run at a time - although it runs very, very quickly. In order to allow many people access to the same data it is advisable to put the data onto a distributed system. By using a distributed system such as a client-server network, people are able to access the same data at the same time, and they run the programs on their local machines, often right at their desks.

It would be reasonable to update an accessible copy of the database on a monthly, or bi-weekly basis. Once the technicalities involved in the transfer have been worked out, much of the transfer can be automated and done with little difficulty. This does create the problem that exists around dynamic data such as ages and lengths of stay, but if the person accessing the data are aware of its limitations, these are easy to deal with.

This suggests the next obstacle, that is, in order to make reliable use of the data you have to understand its structure and limitations. The person working with the data must understand what tables exist, what they contain and how they relate to each other. For example, if during the conversion to the distributed system the program does not transfer birthdates, but rather ages, then the planner must know how often the system is updated and realize that the ages will be off by that amount, i.e., a person might actually be a month older than the database says they are. They must also be aware that people move in and out of the housing units all the time, not just at the ends of months. However, if someone is

doing long-range planning, minor discrepancies such as these should not be an issue, as long as the planner is aware that they exist.

It is also important for the planner to know how reliable the data are. She should know what error-checking mechanisms were used during data input and how much flexibility the person who entered the data had. It is often useful for people using the data to begin with simple queries to ascertain the status of the data. Some fields may not be used and some may not be used much. The person using the data should understand what a blank field means. In some cases it may indicate that the data were not collected from the resident, while in others it might reflect a fact. For example in a field labeled "EMPLOYER" a blank may indicate that the person is not employed. However if there is an additional field "INCOME TYPE" indicating that there is an earned income then the person may indeed be employed nonetheless. Someone familiar with how the data were entered would know if one was an error or an oversight and how likely are similar errors to occur. Another possible source of confusion would be if the column "MOVE OUT DATE" did not have a date in it thereby indicating that there is a current resident, while the column "CURRENT CLIENT" might not have a resident ID in it indicating that there is no current resident. These sorts of discrepancies should be examined and the planner should know which data are more reliable and why, when making her conclusions.

The planner must also be familiar with what the data mean and how they are organized. Some questions result in different answers when asked in different manners. Planners must be familiar enough with the data to know when a question will produce different results when asked differently and why. There are many different questions that planners may ask, and they all require a knowledge of what data are available and where.

There is often a great deal of accumulated information within housing agencies. Planners want to be able to figure out the best ways to view these data. Planners need to be able to understand what groupings of data make the most sense and what questions are worth trying to find the answers to. Hopefully the agency has all needed information stored in an easy-to-access relational database. However, this is not always the case. Often agencies cut down on the number of variables they store to save space, and they also tend to off-load old data onto tapes. Some agencies keep their data on proprietary database machines with limited tools to access the data, which necessitates complex routines to transfer the data to a different format. Sometimes the data exist but they are at a different

agency, most likely stored in a different format. These issues of data format and location make extracting information from the data more difficult than just having the right tools and knowing how to use them.

Some of the more complicated issues involve the storage formats of data, the difficulties involved when one brings together data from different agencies, with time series questions and with mapping the data. We addressed all of these issues when working with the data from the BHA.

For the BHA project, we dealt with data from the Boston Assessing Department via the Boston Redevelopment Authority (BRA), the census, and the Computer Resource Lab (CRL) at MIT in addition to data from the BHA. I will focus my discussion on the data from the BHA, but I will also describe some of the other data as well.

BRA and Census Data

The data from the BRA were parcel level data originally from the Boston Assessing Department with parcel IDs as well as block, ward and precinct codes and street addresses for the area surrounding Mission Main. These data were then georeferenced to maps of the surrounding areas by using an addressmatching program within Arc/Info. We also had parcel boundary maps in DXF format that were matched with the data and maps of the surrounding streets and demographic data from the 1990 US Census for block groups. These efforts illustrate the variety of obstacles that must be overcome when working with data from more than one source, as well as the opportunities afforded when juxtaposing theme rich data sets.

BHA Data

The BHA first transferred their records to electronic format in 1985. For the most part, the data they currently have on-line is complete from 1985 to the present with some information going back further. They keep information on-line regarding family composition, income, rent and tenure in Boston housing developments. They also keep information on the individual units, when people move in and out of the units, state of repair and the reason why the last person moved out.

BHA Data Format

The data at the BHA are stored in a hierarchical 'pick' database. This is difficult to describe in words, but essentially, the tenant data are stored in such a way that there is a master file of tenants and a master file of units which can be cross-referenced using unique housing unit and household IDs. Each master file has several sub-files. The household master file includes information about each household and each household record has a pointer to additional information about the household and each member of the household. In this member file, each record has a pointer to additional employment and income information, if there is any, in an employment file that holds employer information for the member of the family. In the household file, each record has a pointer to a unit number file of move-in and move-out dates for the unit indicating who moved in and out when. There is also a master housing unit file that has all relevant information on each housing unit. This file has a pointer to the record in the member file for the current client as well as a pointer to the unit file of move-in and move-out dates.

The information we received from the BHA is only for those people who live or have lived in Mission Main. We received information on 2560 households and 2515 rows of data on units. Currently 549 of these households reside at Mission Main. The data we received is summarized in Table 4-1.

Oracle Name	Data	BHA table name	Number rows
CUNITSMH	'housing unit' file	TAFUNIV	951
CHEMFMH	'head-of-household'	HEFHEMF	1418
CHEHSMH1	'family member' file	HEFHEHS	4206

Table 4-1 Basic tables used in research

In order for the BHA to give us these data it needed to be converted to a flat-file format from the hierarchic format that the BHA uses. We received a client file that had a household index number through which we could relate which records in the family member file corresponded to each client record. The member file includes income information for each member of the family and an extra row for employment information for the head of household, unless this row is used for a spouse. The employer record file also uses these household index numbers as well as a file sequence number to relate which employer matches with which family member. Each client record included the unit number and their most recent move-in and move-out dates. We also received a unit file that included a row

for each time a household moved into a unit and the household index number for the household. Currently occupied units are indicated by having no move-out date and a client ID greater than zero in the field "CURRCLIE" or current client.

The data were received from the BHA on 3 diskettes in Lotus *.wk1 format. These were then read into Excel for processing as is shown in Table 5-3. The software at the BHA provided other tools to facilitate moving data to relational tables but the software was recently installed at the BHA and the tools were not yet functioning.

There were several steps that were needed in order to get the data into a format that could be used for the class. The details of these steps are listed in Appendix A to give the reader an idea of the difficulties of transitioning from one format to another.

Below I list the three main tables that the class worked with and that I will be using in this thesis for sample data. I also give a listing of some of the data in each table.

CUNITSMH	Characteristics of each Mission Hill housing unit, 951 units are described and 549 have current clients.
CHEMFMH	Characteristics of each head-of-household that has 'ever' lived in a Mission Hill unit. (One row for each of 1418 households)
CFAMILY	Characteristics of each member of each household, 4206 family members (including the head of household) are described.

The following is the data dictionary for each of these tables and a sample of the data from each table. This information for the subsidiary tables can be found in Appendix B.

The CUNITSMH file holds information about the individual units. It holds information that is specific to units, such as its address and the number of bedrooms. The one field in this file that changes regularly is CURRCLIE. This field represents the "Current Client" that is living in the apartment. If the apartment is empty this field value is zero. The other fields are for the most part static for each unit. A sample of the data follows the data dictionary.

SQL>describe CUNITSMH;

Name	*Format*	*Type*
BHASEQ	NUMBER(38)	sequence number of original BHA data
UINDEX	VARCHAR2(6))	BHA housing unit ID
CURRCLIE	NUMBER(38)	BHA household ID of current resident
HOUSNO	NUMBER(38)	street number
STRENAME	VARCHAR2(10)	street name
APARNO	NUMBER(38)	apartment number
BUILNO	NUMBER(38)	building number
BEDR	NUMBER(38)	bedrooms in unit
SITECD	NUMBER(38)	
QTRB	NUMBER(38)	
UNITSTAT	VARCHAR2(15)	(breakthrough, funded for mode)
UNITTYPE	NUMBER(38)	code
FLOOR	NUMBER(38)	floor unit is on
VACREASON	VARCHAR2(12)	why unit is vacant code

BHA SEQ	UINDEX	CURR CLIE	HOUS NO	STRE NAME	APA RNO	BUIL NO	BED R	SITECD	QTR B	UNIT STAT	UNIT TYPE	FLOOR	VAC REASON
1	MH0001	60378	691	PARKER ST	1	1	6	103	1		0	1	UNKNOWN
3	MH0002	0	691	PARKER ST	2	1	3	103	1	BREAK THROUGH	0	1	
4	MH0003	11662	691	PARKER ST	3	1	3	103	3		0	2	
69	MH0028	54642	8	RACINE CT	28	0	3	103	3		0	3	DECEASED
71	MH0029	68424	8	RACINE CT	29	0	2	103	3		0	3	
74	MH0030	19260	8	RACINE CT	30	0	3	103	3		0	3	
78	MH0031	62702	7	RACINE	31	0	3	103	3		0	1	UNKNOWN

CT

The CHEMFMH table holds information about each household. It includes information useful in determining rent and where the family should be placed. It includes the number of bedrooms needed, disabilities, and information about the head of household. Each row has a unique HHINDEX which is the unique identifier used for the household. This is the number that is found in the CURRCLIE field above, and it is the number that is used in the family table (CFAMILY) to identify which household each person in that table belongs to.

SQL> describe CHEMFMH;

Name	*Format*	*Type*
BHASEQ	NUMBER(38)	sequence number of original BHA data
HHINDEX	NUMBER(38)	BHA household ID
STATUS	CHAR(9)	
BEDR	NUMBER(38)	# bedrooms needed
ELIG	VARCHAR2(4)	
DTYPE	VARCHAR2(2)	disability code if relevant
SEX	VARCHAR2(3)	sex of Head of Household
BIRTDATE	CHAR(11)	birthdate
AGE	NUMBER	age to 2 decimal places
CONDOFHO	CHAR(13)	condition of previous housing? (Homeless, substandard, condemned)
MARISTAT	CHAR(12)	(married, widowed, divorced, single, unknown or blank)
RACE	CHAR(11)	(white, black, Spanish)
RENTPCT	NUMBER(38)	percent of income paid to rent
REEXREVI	NUMBER(38)	
NEWRENT	NUMBER(38)	rent they pay
FAMICOUN	NUMBER(38)	number of persons in family?

BHA SEQ	HHIN DEX	STA TUS	BEDR	ELIG	DTYP	SEX	BIRT DATE	AGE	CON DOF HO	MARI STAT	RACE	RENT PCT	REEX REVI	NEW RENT	FAMI COUN
1	75	WV	1			M	07/17/08	86.8		MARRIED	WHITE	30	3	128	1
2	676	CT	3	1		F	04/26/26	69.0		WIDOWE D	BLACK	30	4	685	3
6	692	CV	3			F	04/12/20	75.1			BLACK	30	3	323	5
61	3283	CV	1			F	06/09/41	53.9		SINGLE	BLACK	30	3	163	1
63	3297	WT	1			F	04/15/35	60.1		SINGLE	BLACK	30	5	26	1
65	3307	CV	2			M	02/03/30	65.2		MARRIED	BLACK	30	5	179	3

The CFAMILY table holds information about each person in BHA housing. It contains information that is unique to individuals such as age, sex and race as well as an entry called HHINDEX which indicates which household the person belongs to. This file can besome what confusing because in addition to one row for each family, there is a second row for each head of household that is either used for additional information about that person or for information about his or her spouse, if there is one.

SQL> describe CFAMILY;

Name	*Format*	*Type*
BHASEQ	NUMBER(38)	sequence number of original BHA data
HHINDEX	NUMBER(38)	BHA household ID
MEM	NUMBER(38)	membercode, 1st, 2nd or 3rd person...
RELATIONSH	VARCHAR2(10)	relation to head of household
SEXARRA	VARCHAR2(2)	sex of member
BIRTDATE	VARCHAR2(9)	birthdate
FAMISTAT	VARCHAR2(8)	status of member (elderly, disabled, employed, family, minor, handicap, at home)
RACE	VARCHAR2(8)	race
ETHNICITY	VARCHAR2(10)	(non-hispanic, Hispanic, blank)
NAT	VARCHAR2(3)	(1, 2)

HASEQ	HHINDEX	MEM	RELATIONSH	SEXARRA	BIRTDATE	AGE	FAMISTAT	RACE	ETHNICITY	NAT
1	75	1	HEAD OF FA	M	07/17/08	86.8	ELDERLY	WHITE	NON-HISPAN	
3	676	1	HEAD OF FA	F	04/26/26	69.0	DISABLED	BLACK	NON-HISPAN	
5	676	3	DAUGHTER	F	07/09/56	38.8	EMPLOYED	BLACK		
6	676	4	SON	M	11/18/47	47.4	FAMILY	BLACK		
99	3283	1	HEAD OF FA	F	06/09/41	53.9	EMPLOYED	BLACK	NON-HISPAN	
101	3297	1	HEAD OF FA	F	04/15/35	60.1	EMPLOYED	BLACK	NON-HISPAN	
103	3307	1	HEAD OF FA	M	02/03/30	65.2	AT HOME	BLACK	NON-HISPAN	
104	3307	2	SPOUSE/CO-	F	08/20/32	62.7	AT HOME	BLACK		
105	3307	3	SON	M	12/06/65	29.4	FAMILY	BLACK		

As we can see there is a great deal of information in these tables. Much of this is kept for administrative reasons and many of the field names are for historical reasons. Planners often need to see only subsets of the full tables, and in fact, these are mainly subsets of the full tables used by the BHA.

These three tables provide information about the units, the households and the families. These are the master tables. There are other subsidiary tables that hold additional information about individual's employment, employers, family income and move-in and move-out dates per family and per unit. More information about these tables can be found in Appendix B.

Answering Planners' Questions

I will now attempt to illustrate some of the questions presented above, and show the process of answering them using SQL and a relational database.

The section "SQL and Relational Databases" explains the concepts of a relational database and the basics of the SQL language. I will now attempt to build on this knowledge to answer the typical questions raised above.

First level queries: Tenants and Units

To illustrate the language and how it works, I will begin with Type I questions. These questions can be answered using virtually any system or procedure a housing agency might employ, I include them here to illustrate the basics of SQL and for completeness.

We first ask "How many current tenants do we have?" We take a look at the file that indicates whether or not clients are current, and count how many are current. For example:

```
SELECT count(*)
FROM CUNITSMH
WHERE CURRCLIE > 0;
```

to get a count of how many units currently have households in them, and thereby, the number of current tenants. "Count(*)" indicates that the query should count the number of rows that are selected with the rest of this command. "**FROM** CUNITSMH" indicates the table that the query should use. The last line, "**WHERE** CURRCLIE > 0;" tells the query which rows should be selected. From a previous examination of the data we know that when there is someone living in the unit the field "CURRCLIE" has as its value the household index number of the current tenant, which is known in other tables as "HHINDEX". When the unit is empty this field holds a value of 0. Running this query tells us that there are 549 current households.

If we wanted a listing of all the addresses of all current clients, we could say:

```
SELECT CURRCLIE, HOUSNO, STRENAME, APARNO
FROM CUNITSMH
WHERE CURRCLIE > 0;
```

Note for this query I chose the names available to us, therefore we only retrieve the household ID. This query is fairly straightforward, logical,

CURRCLIE	HOUSNO	STRENAME	APARNO
60378	691	PARKER ST	1
11662	691	PARKER ST	3
26884	691	PARKER ST	4
16524	691	PARKER ST	5
19601	691	PARKER ST	6
49570	685	PARKER ST	7

and practically in plain English. It would not be difficult for someone to learn or to remember. These automation queries take only the most basic understanding of the language and the data.

Second Level Queries: Race

The next set of questions is only conceptually more difficult, while the actual queries are the same level of complexity. These questions also raise the issue that the data might not be collected consistently, if at all. Before attempting to ask the question, it is good practice to look at a listing of the data to see how consistently the variables are filled in, and to see if there is any consistent reason for when they are not filled in. For example, race is never filled in when the head of household is a minor, or birthdate is not present when the head of household is a senior citizen.

I will take a look at two of the race questions I suggested earlier. To say "How many of our tenants are black?" We might formulate a query such as the following:

```
SELECT RACE, count(*)
FROM CHEMFMH
WHERE RACE = "BLACK"
GROUP BY race;
```

RACE	COUNT(*)
BLACK	572

Now if we were interested in all races we would look at this query:

```
SELECT RACE, count(*)
FROM CHEMFMH
GROUP BY RACE;
```

RACE	COUNT(*)
AM IND	2
ASN&PAC	15
BLACK	572
OTHER	5
SPANISH	766
UNKNOWN	1
WHITE	57

However, we should be aware that our data have some intricacies. Because of the consistency of the result, we can tell that race is entered in a standardized way. By this I mean that the race is never misspelled and is always abbreviated the same way. For example, we have 15 entries of ASN&PAC, referring to Asian or Pacific. If this was not standardized, we would probably have entries of "ASIAN", "JAPANESE", "THAI", "A&P", etc. It is unlikely that without a pick-list or some other way of limiting entry to the database the results we get would be so standardized. However, this is no guarantee that

race is always entered, we might want to verify this by looking for where this column is null. If only a small proportion of the records actually have the race indicated, then it may not be statistically meaningful for our purposes. To do this we might want to do a quick query such as:

```
SELECT BHASEQ, RACE
FROM CHEMFMH;
```

Which would give us a listing of the columns of BHASEQ and RACE for all the records. We could just scan this visually to see if there are any inconsistencies or missing values. However, this could be very tedious and time-consuming for a very large database In that case we may prefer to do a couple of queries that would check this for us. Our query that grouped by races has already indicated that any entries are consistent, however, it would not count rows where RACE was NULL. Such a query might be:

```
SELECT count(*)
FROM CHEMFMH
WHERE RACE IS NULL;
```

COUNT(*)
0

This indicates that race is always filled in and it is standardized. A planner who is familiar with the program through which this data are entered, and policy at the agency would probably already know how reliable the various fields are and would not need to do these investigation queries.

We should recognize that this table doesn't only contain current residents, so if we want to know what percent of the entire population is of what race, then we'd look at that percent compared to the percent of current clients. We could get this information by asking how many clients are in our database, and doing simple division to get percentages:

```
SELECT count(*) FROM CHEMFMH;
```

COUNT(*)
1418

To look at only current clients involves combining the units table with the household table for all our queries. We would do this with a query called a join. The query would be done with a statement such as.

```
SELECT RACE, count(*)
FROM CHEMFMH c1, CUNITSMH c2,
WHERE c1.HHINDEX = c2.CURRCLIE
      AND c2.CURRCLIE > 0
GROUP BY RACE;
```

RACE	COUNT(*)
ASN&PAC	7
BLACK	196
SPANISH	330
WHITE	16

This query would select only the rows where the value of CURRCLIE is greater than zero, which we have already established means that it is a current tenant. In this way, only the current tenants will be counted by the count(*) command.

We should note that since this table is selecting from the table CHEMFMH, we are actually counting households, not individuals of each race. If we were to count individuals, we would need to reference the CFAMILY table and we would obtain different results.

These questions of race would be useful to a planner in understanding the ethnic composition within the housing development, and between the development and its neighbors. Federal law also has several regulations regarding race and discrimination in public housing, thereby making the review of racial breakdowns important in annual reports. Studies of race are also useful in determining how the population of the development is changing with time, and if there are any other demographic factors that seem to be encountered prevalently with race.

The results of our queries show us that this development has a population of 2/3 Spanish households and 1/3 black households. Quick comparison of the origianl query and the new one for current tenants only, indcates that the split hasn't changed dramatically over the past few years. We also see that half of all the Asian & Pacific residents that have ever lived in Mission Main live there currently, so there's a good chance that that segment, albeit a small one, is growing. These race questions have all been straightforward, and are examples of the first and second level questions.

We see how these questions are answered fairly easily when the planner has access to the data, and a basic knowledge of the data structure and tools used to query the data. In a typical housing authority, a planner would need to carefully formulate the questions she would want answered in advance, and submit them to the MIS department. To obtain the same answers as we did she would need to ask, "How reliable are our data about race?", "What is the racial breakdown of all the households in the database?" "What is the racial breakdown of current residents?" "How many total residents are there?" "Is there a race associated with every household?" The planner would need to know the answers to all of these questions in order to defend any decisions she based on the results of her data, but it might not occur to her to ask all of them the first time. By having access to the data herself, the planner does not need to formulate all of her questions in advance, and then wait for

results, but can rather think of additional questions while she works. However, these are still fairly simple and straightforward questions.

Third Level Questions: Employment

As we suggested earlier, the third type contains more complex queries, such as examining tenant employment. Here we look at that issue using the BHA data. We use the data from the table CINCOME to determine what percentages of the current population are gainfully employed, on welfare, or receiving some other sort of income.

This table, CINCOME, contains data indicating the income source for each member of each household in Mission Main, however this data are not standardized. That is, the entry for income source was typed in manually, therefore there are misspellings, varying abbreviations, and different words used for the same things. For instance, to indicate "gainful employment" the following entries were used: "EMPLOYED", "GAIN EMPL", "GAINFUL EM", "PART-TIME". An entry of any of those items indicates that the person should be listed as "EARNED." Therefore, to group these, we made a lookup table that first standardized spelling, and then grouped all the possible entries into "EARNED", "TRANSFER/INS", "WELFARE", or "UNKNOWN". Using this lookup table and the CINCOME table, we were able to select all the individuals that were gainfully employed. This table also gave us the individual's HHINDEX number, which identified them with a unique household. The strategy of lookup tables is discussed more thoroughly in Appendix A.

This strategy involves knowing a bit more about how the data are structured than was necessary in some of our earlier queries. It also involves more preparation of the data. After creating this lookup table, we can now, however, go ahead with our queries about income source, and perhaps look at this information in conjunction with other data that are available to us.

However, we know from before that not all the tenants in our database are current tenants, therefore only some of the tenants we have selected so far interest us. To find out which tenants are current, we must check if their HHINDEX shows up in the "CURRCLIE" field of the units table. Perhaps when doing the race query earlier we realized that we might want to look at just the current clients again, and in that case we would have saved the current tenants as a temporary table called CURRTEN. More

information about creating temporary tables can be found in Appendix A. If the HHINDEX shows up in that table, then the household is current.

If we do a query, and ask for data from both the CINCOME table and the CURRTEN table, only rows that show up in both places will appear in our results[3] . In this way, we find our answer by first relating the employment lookup table and the income table, and selecting only the rows where the income source is "EARNED". Then we relate these rows with our table CURRTEN. The result of a relate of those two will only show us the rows where the two fields match, thereby indicating current clients. However, there may be multiple entries per household, if more than one household member earns income, therefore we should group our results by HHINDEX to give us an accurate count of the number of households with earned income. By counting the total number of rows in the table CURRTEN, we can determine how many households are in our study group and then simple division gives us the percentage of those households that have at least one member of the family employed.

Once we have the percentage of current households with at least one member employed, we may wish to look at the same numbers for all the households in the database. This would give us an indication of whether or not employment rates for people in our development are increasing. We may also wish to find out how many households have more than one family member employed. If we have saved our query from before, where we grouped our results by HHINDEX, we can run it again, this time including a request for the column count(*). This would tell us how many people are employed in each household. An easy way to see how common it is for households to have more than one person employed would be to order by that column, count(*).

These, and other similar queries, are all fairly straightforward once the basic organization of the data is done. If a planner is doing these things herself, she knows what is available, and she is able to build on her results. If she were doing these queries through the MIS department, it might not occur to her that some families may have multiple sources of income, because she may not know how the data are structured. She might not realize that she wants that information until after she receives the results of her original query. She might then submit a request for this new information, but the MIS department may have

[3] That is, unless we do something called an "outer join" which for the purposes of this thesis we will not deal with.

already discarded the temporary tables and the queries used to find the data, thereby forcing them to recreate all that information. It is much more efficient when the planner can do the queries herself.

These examples can become even more complex and involved, but with this basic understanding of the tools and the data, a planner should be able to handle these issues. Some of the reasons behind allowing the planner direct access to the data include the speed and efficiency with which the results can be obtained, and time saved by the MIS person not having to explain what is or isn't possible. The planner would have a better idea of what temporary tables or commands she might want to keep, in order to do further study later, and this creates increased opportunity for ad hoc questions and answers.

Fourth Level Questions: Tenure

A fourth level type of question might look at tenure, for example, how long have various families lived within Mission Main. This discussion is more complex and technical than the previous sections. Examination of tenure questions is generally more complicated than other queries, because of the added dimension of time. Change over time is difficult to represent in a database. For instance, if a family member moves out, do you leave that persons information there and add a column for personal move-out date? Do you continue to compile information about that person? Generally you cannot, even if you wish to, because the family will stop giving you information about that person. What if there is a new family member, how do you account for that? If a child is born, then obviously a birth date will indicate when that person joined the family. If I wanted to take a survey of all the single-parent households five years ago, and compare them to those today I would experience these difficulties. For example, if someone got married between five years ago and today, then from the data we have, I would not know that they were a single-headed household five years ago. If they still live in the development, then I would only have indication that they are married. If they moved out before getting married, I would know that they were single five years ago, but I would not be aware that they are now married. This is just one example of the types of difficulties one might run into when attempting to analyze data that change over time.

Most of the following queries examining tenure are done with two tables, CUMOVES and CHMOVES. These tables contain information about move-in/out dates for the units and the households respectively. The data dictionairy and formats of these tables are listed in Table 4-1and Table 4-2.

SQL> describe CHMOVES;

Name	Format	Type
BHASEQ	NUMBER(38)	sequence number of original BHA data
HHINDEX	NUMBER(38)	BHA household ID
UNITSTACK	CHAR(11)	BHA housing unit ID
MOVEIN	CHAR(10)	move-in date
MOVEOUT	CHAR(10)	move-out date
DATEIN	NUMBER	Julian date (days since 12/31/4713)
DATEOUT	NUMBER	Julian date (days since 12/31/4713)
TENURE	NUMBER	months in unit (as of 41995)
VACAREAS	CHAR(10)	vacancy reason

BHASEQ	HH INDEX	UNIT STACK	MOVE IN	MOVE OUT	DATEIN	DATE OUT	TENU RE	VAC REA
1	75	MH0028	03/14/41	06/30/89	2430068	2447708	588.0	D
2	676	MX5403	06/29/89	2447707	2449827	70.7		
3	676	MX5803	06/02/89	06/28/89	2447680	2447706	0.9	U
4	676	MH0276	09/05/86	06/01/89	2446679	2447679	33.3	U
5	676	MX2062	09/01/54	09/04/86	2434987	2446678	389.7	U
6	692	MH0449	09/23/86	06/25/87	2446697	2446972	9.2	U
7	692	MX2056	10/01/54	09/22/86	2435017	2446696	389.3	U

Table 4-1 Data Dictionary and Structure of CHMOVES

SQL>describe CUMOVES;

Name	Format	Type
BHASEQ	NUMBER(38)	sequence number of original BHA data
UINDEX	VARCHAR2(6)	BHA housing unit ID
CURRCLIE	NUMBER(38)	BHA household ID of current resident
HHINDEX	NUMBER(38)	BHA household ID
MOVEIN	VARCHAR2(8)	move-in date
MOVEOUT	VARCHAR2(8)	move-out date (blank if current resident)
DATEIN	NUMBER	Julian date (days since 12/31/4713)
DATEOUT	NUMBER	Julian date (days since 12/31/4713)
TENURE	NUMBER	months in unit (as of 4/19/95)

BHASEQ	UINDEX	CURR CLIE	HHIN DEX	MOVE IN	MOVE OUT	DATEIN	DATE OUT	TENU (mont
1	MH0001	60378	60378	01/09/92		2448631	2449827	39.9
2	MH0001	0	11477	02/01/76	10/09/91	2442810	2448539	191.0
3	MH0002	0	4082	10/15/68	02/28/75	2440145	2442472	77.6
5	MH0003	0	18144	07/01/81	06/01/85	2444787	2446218	47.7
6	MH0003	0	2741	08/03/65		2438976	2449827	361.7
7	MH0003	0	0	05/15/77		2443279		.0
8	MH0004	26884	26884	12/22/92	2448979	2449827	28.3	

Table 4-2 Data Dictionary and Structure of CUMOVES

The CUMOVES table is unit based. It includes a record for each time a household moved in or out of a unit, the most recent client and move-in and out dates, and a reference to the current client if the household is still in the unit. The CHMOVES table has the same information from the household's point of view. It has one record for each time a household moved in or out of an apartment, the unit number, the move-in and out dates and the reason they left. The move-in/out dates were calculated using the Julian date[4] in order to facilitate further calculations. The dates have been set so that the move-out date for current clients is the day the data were transferred from the BHA to MIT (April 19, 1995).

Setting the move-out date illustrates one of the difficulties of working with time sensitive data. If we were to do queries such as the following with data in an active database, instead of a snapshot such as this one, the move-out date for current clients would have to change every day. Either current clients would have to be excluded from time studies, or each day a new move-out date would need to be calculated. Because of these potentially different move-out dates, if the same queries were run on different days,

[4] The Julian date is a number that indicates the number of days since a date in the past, around 4000BC.

different results would be received, making it difficult to compare the results. By taking a snapshot of the data we avoid these problems, but it adds the complexity of removing a large portion of the data from the original database, including the data translations that are discussed in the Appendix, and the data are not as current as one might hope.

As we begin our discussion of tenure, I first want to error check the tenure data to see how reliable they are. I found two types of possible errors to watch out for. In the CUMOVES table, if there is no current client, CURRCLIE is set to 0 and MOVEOUT is left blank. Therefore I checked to see if it ever occurred that when CURRCLIE is greater than 0, MOVEOUT is NOT NULL. To do this, I used Query 4-1. A sample of the results are listed in Query Results 4-1. 23 rows were returned, 14 of which had negative tenures. Where the tenure is negative I assume that the move-out date is in error. In the other rows it is not clear if the move-out date is the error or if the CURRCLIE variable has simply not been set back to 0 to indicate that someone has moved out.

```
SELECT * FROM CUMOVES
WHERE CURRCLIE > 0
      AND MOVEOUT IS NOT NULL;
```

Query 4-1 Checking for where there is a current client and a move-out date.

UINDEX	CURR CLIE	HH INDEX	MOVEIN	MOVE OUT	DATEIN	DATE OUT	TEN URE
MH0010	16371	17380	03/02/81	05/25/76	2444666	2442924	-58.1
MH0155	9533	9533	03/02/85	03/31/76	2446127	2442869	-108.6
MH0176	19731	19731	01/11/85	01/31/83	2446077	2445366	-23.7
MH0263	10895	10895	06/01/85	05/15/76	2446218	2442914	-110.1
MH0473	5313	8851	04/01/74	11/19/76	2442139	2443102	32.1
MH0477	5381	6777	08/16/72	07/18/77	2441546	2443343	59.9

Query Results 4-1 Checking for where there is a current client and a move-out date.

Another interesting selection is to look at where the CURRCLIE does not match the HHINDEX. HHINDEX is supposed to reflect the occupant that moved in and out on the MOVEIN and MOVEOUT dates. If there is no MOVEOUT date, this would always be the current occupant or CURRCLIE. We found that there were 11 instances where these did not match. It is interesting to note that these instances were all reflected in the previous query, because they also had MOVEOUT dates, when they should not have. Therefore, these do not reflect 11 additional errors, but that the rows indicated in Query 4-1 are additionally suspect. The query and its results are listed in Query 4-2 and Query Results 4-2.

```
SELECT * FROM CUMOVES
WHERE CURRCLIE != HHINDEX
      AND CURRCLIE > 0;
```

Query 4-2 Selecting where the current client does not equal the most recent client

UINDEX	CURR CLIE	HHIN DEX	MOVEIN	MOVEO UT	DATEIN	DATEOU T	TENU RE
MH0010	16371	17380	03/02/81	05/25/76	2444666	2442924	-58.1
MH0473	5313	8851	04/01/74	11/19/76	2442139	2443102	32.1
MH0477	5381	6777	08/16/72	07/18/77	2441546	2443343	
MH0488	3435	15292	04/16/79	07/22/81	2443980	2444808	27.6
MH0490	17970	19894	10/01/82	09/09/83	2445244	2445587	11.4

Query Results 4-2 Where the current client does not equal the most recent client

This makes me curious as to how often MOVEOUT, or DATEOUT is incorrect. This would be most obvious by looking at where the tenure is negative (although this could of course indicate that MOVEIN is incorrect as well). Interestingly here we found 97 rows that matched this criteria. This information was found with Query 4-3.

```
SELECT * FROM CUMOVES
WHERE TENURE < 0;
```

Query 4-3 Selecting for where tenure is negative or zero

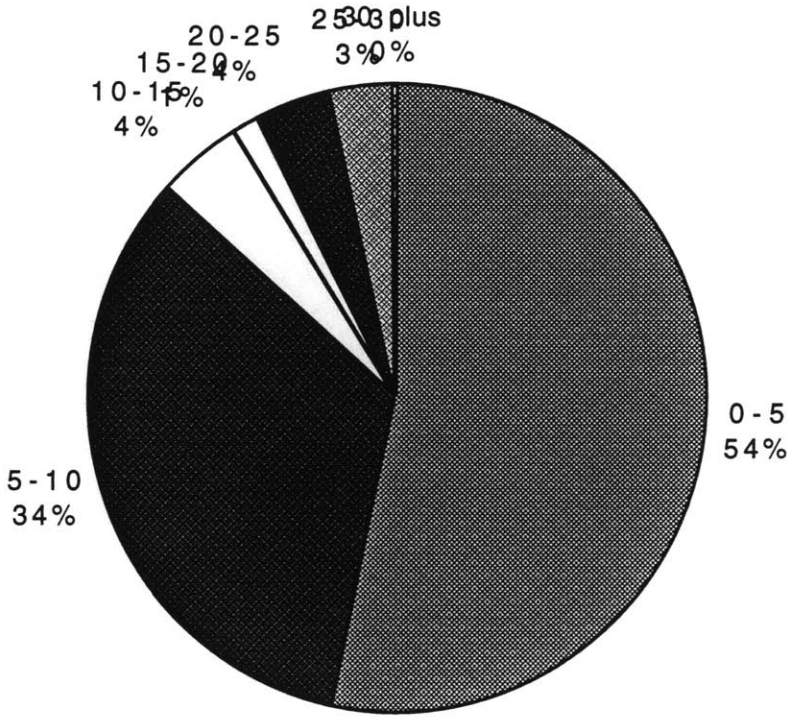
From these queries I get the impression that my data are pretty good, but not perfect. It appears to be reliable enough to do research-type queries on. Now that I have a feel for my data on tenure, I will begin to examine what information I can find in it. One item that we might want to take a look at is how long residents have been in Mission Main and how long they have been in their current apartment. We might wonder if the current tenants have spent most of their time in their current apartment or if they have moved around within Mission Main. To do this we might first take our table of movein/outs for units and pull out a subset of this table that only represents the records of those households that still live in Mission Main. To do this, we would compare the HHINDEX for each record in the CUMOVES table with the CURRCLIE field in the CUNITSMH table. If a HHINDEX shows up in the CURRCLIE field in CUNITSMH, that indicates that the household currently lives in a unit in Mission Main. We want to leave out the CURRCLIE that are 0,

so we use AND u.CURRCLIE > 0 and we only want those records where the TENURE is positive. Where it is negative there must be some error, so we leave those out of our study. We want a selection that includes not only their current movein/out record, but all the movein/out records of the current clients. To do this, we would use the following query. A subset of the results follows:

```
CREATE TABLE CURRUNITS AS
SELECT m.UINDEX, m.CURRCLIE, HHINDEX, BUILNO, FLOOR, MOVEIN,
       MOVEOUT, DATEIN, DATEOUT, TENURE
FROM CUMOVES m, CUNITSMH u
WHERE m.HHINDEX = u.CURRCLIE
      AND u.CURRCLIE > 0
      AND TENURE > 0;
```

UINDEX	CURR CLIE	HHIN DEX	BUI LNO	FLO OR	MOVE IN	MOVE OUT	DATE IN	DATE OUT	TEN URE
MH0045	1045	1045	2	3	10/23/91		2448553	2449827	42.5
MH0126	2100	2100	5	1	03/05/63		2438094	2449827	391.1
MH0090	2239	2239	3	1	09/15/63		2438288	2449827	384.6
MH1014	2435	2435	38	3	08/01/64		2438609	2449827	373.9
MH0185	0	2535	20	2	12/12/90	11/01/92	2448238	2448928	23.0
MH0551	2535	2535	20	2	11/02/92		2448929	2449827	29.9
MH0672	0	2535	20	2	12/01/64	12/11/90	2438731	2448237	316.9

First we can take a look at a chart of the lengths of time households have spent in their current apartments. We have taken a listing of only the movein/out records for the current tenants and ordered it by tenure. Then we are able to group the households by length of time in the current unit. We see that more than half of the households have been in their current unit less than 5 years and another 34% have been in their current unit only five to ten years. This can be seen in the chart below. Simple calculations tell us that the average time spent in the current unit is six and three-quarters years.

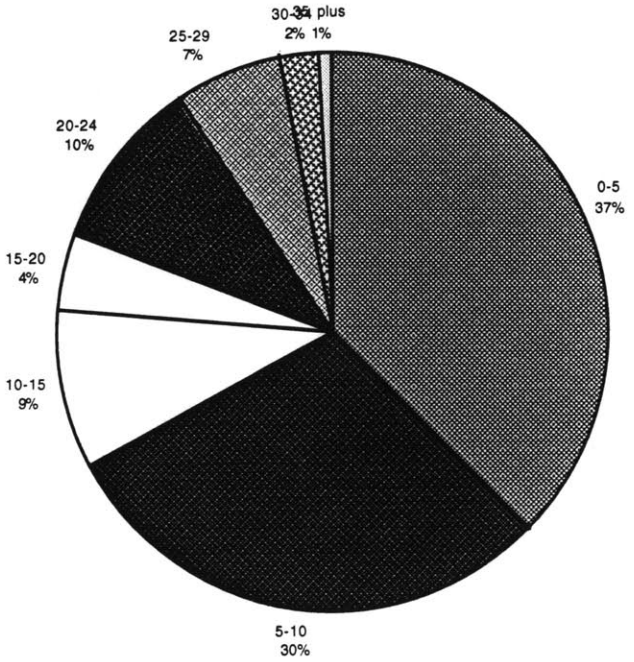


I obtained the numbers in this graph by selecting for only the records related to the unit each household is currently in. Next I look at the total tenures for each household. I do this by selecting for all the records in our table of current households, and then grouping them by household and summing the tenure for each household. This is done with the following query, and the table that follows is a subset of the output of the query.

```
SELECT HHINDEX, count(*) numoves, sum(TENURE) totten
FROM CURRUNITS
GROUP BY HHINDEX
ORDER BY NUMOVES desc;
```

HHINDEX	NUMOVES	TOTTEN
4291	4	318.06
5493	4	405.40
14774	4	279.93
16481	4	182.60
26412	4	118.73
19413	4	156.23
16098	4	186.13
2535	3	369.80

We can then take and with a spreadsheet look at this data in the same sort of chart as before. Doing this we see that 36% of the households have lived in the development less than five years, 29% have lived in the development 5 to 10 years.



It would next be interesting to aggregate this information by building. In that way we might see if the people who have lived in the development the longest all gather in certain buildings. For example, we could ask, what is the average length of tenure in the current unit per each building. For this, we can select for the records with current tenants, and then group by building taking the average tenure for each building. The query that would do this looks like this:

```
SELECT BUILNO, avg(TENURE) avten
FROM CURRUNITS
WHERE CURRCLIE > 0
GROUP BY BUILNO
ORDER BY BUILNO;
```

BUILNO	AVTEN
1	77.67
2	91.29
3	70.69
4	135.17

Although this information is interesting, it's difficult to visualize what it means, in the next section we will map this data to see if there is any pattern to the average tenure in a building. We see that the shortest average tenure is just under 2.5 years in building 20 and the longest is 11.25 years in building 4.

To look at average total tenure per building is a bit more complicated. This is a function of how the data are structured. In this case we want to group by HHINDEX to get total tenure per household and then by BUILNO to get the average tenure per building. One might suggest that you don't need to group by HHINDEX first, because you could just average all the individual tenures. Unfortunately that does not work in this case because we want the tenure to be associated with the last building the family lived in, even if some of that time was spent in different buildings. The idea is to see where the households with the longest total tenures end up. To do this then, we can group by household to get the average tenure per household, as we did earlier, but this time save it to a temporary table, at this point we ignore the BUILNO and do not select it. Perhaps we actually saved it to a temporary table earlier. We can then take this listing and re-associate the households with building numbers by relating it to the CUNITSMH table. However, there may be several entries fore each household. If we relate the HHINDEX from our temporary table to the CURRCLIE field, we will get only the most recent location of the family. We can then take this information and group by BUILNO to get the average total tenure for each building. The queries to do this are listed below.

```
CREATE TABLE TEMP2 AS
SELECT HHINDEX, sum(TENURE) totten
FROM CURRUNITS
GROUP BY HHINDEX;

SELECT BUILNO, avg(TOTTEN) avten
FROM TEMP2 t, CUNITSMH u
WHERE t.HHINDEX = u.CURRCLIE
GROUP BY BUILNO;
```

BUILNO	AVTEN
1	121.83
2	95.32
3	83.98
4	165.17

Although I have only given a partial listing here, the data shows that the shortest average total tenure is 6.8 years in building 27 and the longest average total tenure is 14.75 years in building 26. In the next section we will look at this data spatially to see if there is perhaps any patterns to this.

The fifth category were queries that are sufficiently complex or require special data collection and would require hiring an outside consultant to do the work. Hences, their detailed discussion is beyond the scope of this thesis.

Mapping and Georeferencing

Electronic mapping is a rapidly growing field. Mapping software allows the user to input an address, or a file of addresses, and the software will mark the location on an electronic map. Other information can be added to the map, such as associating census data and census blocks with areas on the map. Then we can ask spatial relation questions, such as, "Is this point in or out of this area," "or how close is this point to an area like this?" These capabilities can be applied to questions from a housing authority.

A planner in a housing authority may wish to ask, "Where are the residents located who have the longest tenancies and what is the surrounding area like, that might account for the these long tenancies?" Planners are now able to ask "Does the proximity of various services have an affect on the length of tenancies in various projects?" "Do the demographics of the surrounding areas have such affects on tenants?" "Where are the apartments most often turned down located? are they near each other? Do the surrounding areas have similarities? If so, what?" Because of the increasing ease of use of mapping tools, and the declining costs of maintaining these data, questions like these can be asked before agencies decide where and how to spend their money. Because these, and other similar questions, can be asked by a manager sitting in his or her office in an afternoon, research and analysis can be more heavily relied upon. There is no need to submit requests for reports in excruciating detail, that will take weeks to be generated, there is no longer a need for a middle man between the manager and the computer. Today's computer can be used to answer many questions simply and relatively easily.

As an example, I have taken some of the database questions on tenure that I asked in the previous section and turned them into maps of Mission Main. As part of our class project, several students digitized the building footprints of Mission Main and turned these into coverages that can be viewed in Arc/Info and ArcView2. Using the ArcView2 SQL-connect function I am able to access the tables in Oracle in the mapping software. This allows me to map the data in our database, once I have grouped them by building in a meaningful manner. Some of the questions that would be interesting from the information we have already generated include, where do the people who have the longest tenure at Mission Main live? Do people who have lived in Mission Main the longest prefer particular buildings? We might also want to look at the average length of tenancy for the current buildings? Are they similar across the development? These questions may give a feel for

which are the "preferred" parts of the development. Examples of the results of these queries can be seen as maps on the following page.

The maps I created for this thesis both look at average tenure per building. I used the same dataset as for my level four queries, that is, households living in Mission Main as of April 19, 1995, known as "current households." I excluded any entries in the database where the tenure for the household's stay in an apartment was listed as being negative. For these maps the households are associated with the unit, and building, they are currently living in. The "Current Tenure" map uses only the length of time spent in the current unit. The "Total Tenure" map associates the family with the current unit, but the tenure is defined as the total time the household has spent in ANY unit in Mission Main. Each building is shaded according to the average for the households currently living in the building. On one map, the buildings are grouped so that there are an equal number of buildings (9-10) in each of the four categories, this is the quartile map. The other map has it such that each category is approximately an equal length of time, about 25 months for Current Tenure and 30 for Total Tenure.

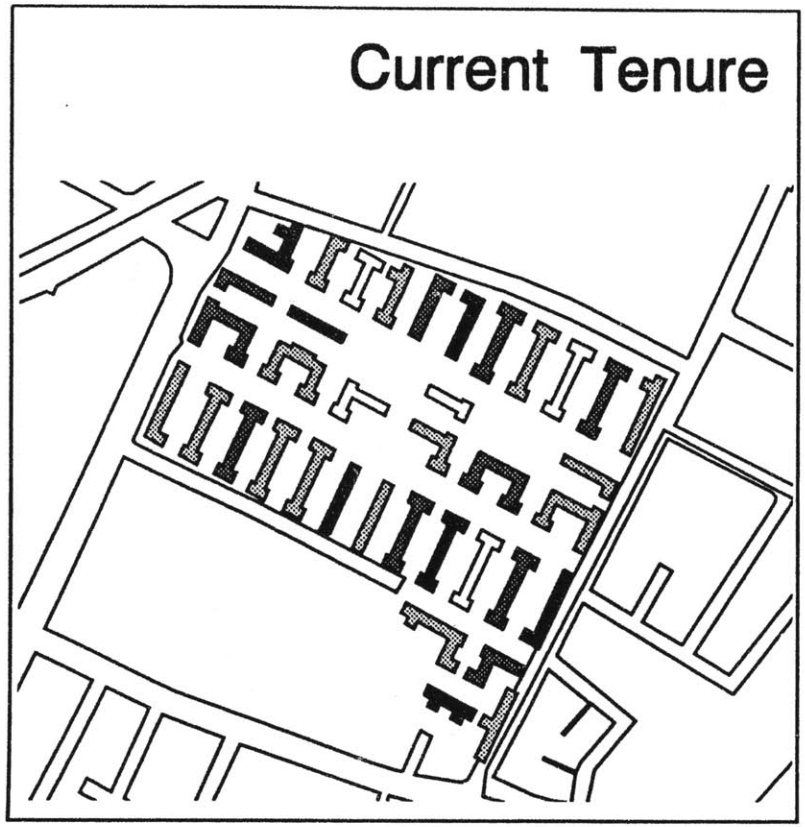
These maps are a good example of how there may be many ways that the maps can be interpreted. The darker shades are the buildings with the longest tenures. We see from the Current Tenure map that, in general, the tenants who have been in Mission Main the longest tend to live in the north-west and south-east corners of the development. This clustering is less evident in the Total Tenure maps. We see that the two maps of Current Tenure have very similar distributions of average tenure. On the Total Tenure maps however, we see that on the equal interval map there are many more buildings in the low end of the scale. That is, there is a skewed distribution with a few families that have been in the development for dozens of years. In a couple of cases, buildings which have the shortest current tenure are in the highest category for total tenure. This may be attributable to the current renovations at Mission Main. Part of these renovations include temporarily relocating tenants among buildings which may result in long term tenants having lived in their current unit for only a short amount of time. This is not because they had been waiting for the current location but because the rennovation schedule forced a number of tenants to move in recently.

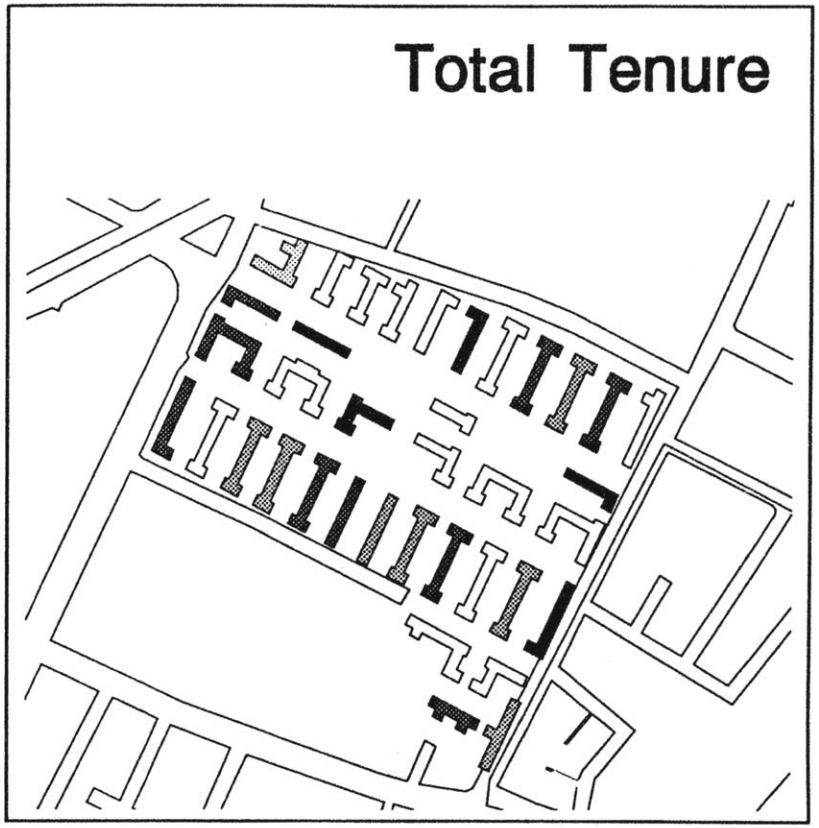
These and other ways of interpreting the maps are good examples of why the planner must be familiar with her data. She should have hypotheses to explore and field experience

to draw upon and be able to understand the results she gets when she creates the maps and submits her queries. With these data for example, I looked at the average, or mean, tenure in each building. The planner might want to consider also looking at the median tenure for each building. The median, or 50th percentile would be another measure of a "typical" situation but would be less influenced by having one of the very long tenure households in a building. She should know which way of viewing the data is more meaningful to her study. She might also consider viewing only the tenure data for a certain household type. For example, she might feel that there is a pattern to where single-headed households with children live. In that case she might want to narrow her data set to be only those households and then re-create the maps.

These are all issues that a planner might consider when deciding how to interpret and present her data. Depending on the study she is doing, or the presentation she is giving, she might make different decisions about how to lay out her data. The following are two ways, out many, many possibilities for examining tenure patterns.

Average Tenure per Building for Current Tenants





Current Tenure (months)
- 29 - 55
- 55 - 80
- 80 - 110
- 110 - 135
- Boston Streets

Current tenure measures the length of time spent in the current unit by each household living in Mission Main. Total tenure measures the total number of months spent by each household in any unit in Mission Main. Categories are rounded equal intervals.

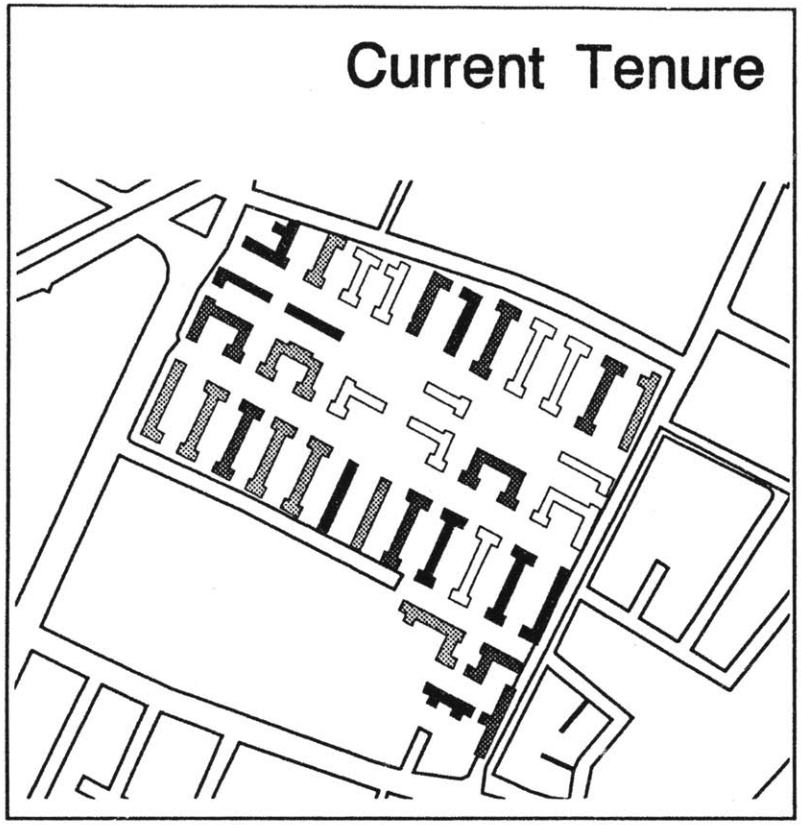
15 0 15 30 Feet

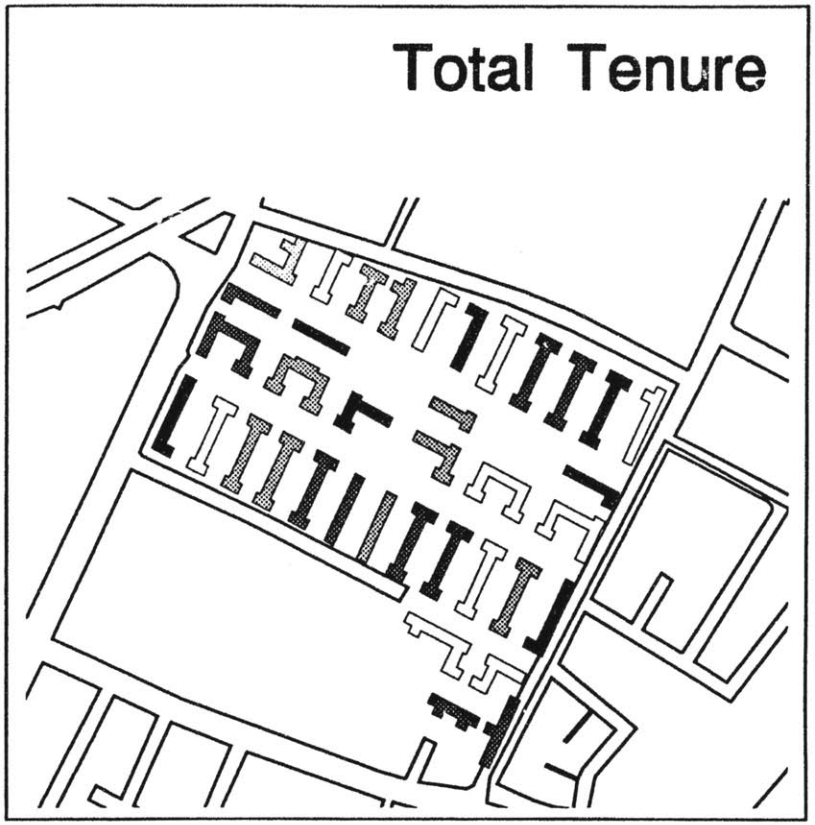
N

Total Tenure (months)
- 80 - 110
- 110 - 130
- 130 - 155
- 155 - 180
- Boston Streets

Alicia Allen, MIT CRL Oct. 1995

Average Tenure per Building for Current Tenants





Current Tenure (months)
- 29 - 63
- 63 - 74
- 74 - 91
- 91 - 135
- Boston Streets

Current tenure measures the length of time spent in the current unit by each household living in Mission Main. Total tenure measures the total number of months spent by each household in any unit in Mission Main. Categories are quartiles.

15 0 15 30 Feet

N

Total Tenure (months)
- 83 - 104
- 104 - 115
- 115 - 145
- 145 - 177
- Boston Streets

Alicia Allen, MIT CRL Oct. 1995

Chapter Five Conclusion

Interpretation

This thesis has discussed housing authorities and their data needs, and has focused in particular on the Boston Housing Authority. It has addressed some of the issues raised by authors in the fields of housing, computers and small business and agency automation. I then examined some of the software packages that are designed particularly for housing authorities. My next step was to present some of the questions planners might want to ask, and group them into types of questions, based on the complexity of answering the question. In answering these questions I pointed out some of the difficulties that might arise and presented some solutions to these difficulties. I then took the results of some of these queries and displayed them geographically on maps.

This thesis asks if it is reasonable to expect planners to perform these queries themselves. There are many advantages to allowing planners direct access to the data and the tools by which they examine the data. There are also drawbacks and obstacles to allowing this access.

Some of the benefits previously discussed include the ability to have the data updated in a more timely and less error prone manner, greater accessibility to a tenant's records by a case manager, and greater flexibility in the types and numbers of queries posed to the data. Direct access allows the planners to obtain answers to their questions in a timely manner. This accessibility reduces the load on the MIS department and allows them to focus more on data management and integrity instead of one-time projects for individuals. Direct access also allows planners to play out scenarios and follow leads, when they have the time and when the ideas occur to them.

There are obstacles to direct access as well, including regulation of whom should have how much access, and the skills and knowledge needed to make responsible use of the data. Planners must know what questions to ask, and have an understanding of the data and what they mean. They must also know how to use the support tools that enable them to answer these questions. There is a large value added by enabling planners to have this understanding and skills. Often one will want to look at the data in many ways, and often subsequent queries are dictated by current results.

This need for the planner to be able to access the data herself is exhibited by my examination of how one would answer the various questions presented. I looked at the data in many different ways, and the next query was suggested by the previous query's results. If I were forced to go through MIS for each view, this examination would probably have taken weeks. I would have spent the first several hours just thinking about what I would want to see, before ever even sending in a request to the programmers to generate my reports. It often takes days, or even weeks to get the resulting tables back, by which point I may have forgotten what I was looking for if I had not made detailed notes earlier. Some of the views I developed only occurred to me because I was able to see the structure of the data and recognize what fields I could relate to each other. Many of my queries looking for errors were also based on this. I would look at the data structure and realize that information could be displayed in different ways. I then looked at these data in both ways to see if there were any discrepancies. For instance, the examples of total tenure for a unit show us how different manners of calculating tenure can result in very different answers.

I then looked at the data from a geographic point of view. These tools are becoming more and more common. Although it is not difficult to ask a GIS person to show you a map that documents tenure, or which shows the apartments with the most turnover in tenants in the last 10 years, often you do not know what questions you will want to ask until you see how the data are formatted. Although some questions may be standard, others may not even occur to one to ask until they see what data are available, and how it might be matched together.

There are other considerations as well, such as what sort of system should you install, and what software package would be most beneficial to your agency. Our discussions reveal that a desktop package that includes relational capabilities is often the best way to go. These packages have the most flexibility in viewing and querying the data, while at the same time provide a structure and format as to how the data are entered. While a command line interface such as I have used is the most flexible and powerful interface, some of the forms packages that act as front ends greatly reduce the amount of learning and knowledge necessary in working with these tools.

Where we go from here?

There are many options and alternatives relating to an information system in a housing agency that I have not gone into here. These include connections to the Internet and bulletin boards where directors of housing agencies can get together to exchange ideas and discuss problems. Management may find Internet access desirable in order to exchange thoughts and ideas with other managers of housing agencies and of low-income housing. There is currently a mailing list called "homeless" on which members of the National Coalition for the Homeless and employees of HUD post announcements of new grants and programs as well as provide a personal contact for more information. This contact is usually more informal than an official phone call and it is usually faster, approximately one day turn-around time. Because HUD does have Internet access, housing agencies are able to FTP documents from HUD, submit reports electronically and keep in regular contact with members of the department.

The Internet also facilitates personal contact between people who otherwise would probably never meet. These personal contacts with the people who make decisions on grants and special programs can help them make more informed decisions. It also allows the directors and staff of smaller agencies to compare difficulties and solutions to specific problems as well as to share ideas and to generally support each other. The Internet is opening up whole new means of communication and opportunities for interaction. In the future everyone will have Internet access and the earlier people get involved the more benefits they will get.

The National Association of Housing and Redevelopment Officials, NAHRO, realized the importance of the Internet and electronic communication several years ago when they developed their own computer information service, NAHRO-NET. This is a dialup service with its own software that allows members to access electronic bulletin boards. These bulletin boards contain summaries of Federal Register notices related to HUD, HUD notices, NAHRO positions and comments on the proposals, legislative happenings, news on housing and community development happenings and places where members can exchange ideas and conference. They also have all of the Journal of Housing articles on-line. This system is still under development and now allows access to electronic mail for its users. (Cowan, 1993)

As the internet and World Wide Web, WWW, continue to develop at its rapidly expanding rate, we can easily imagine a time when all information will be available electronically. Planners will be able to access data for their housing authority remotely, and be able to compare their data with that for other housing authorities. One can imagine an agreement where the Directors of two medium sized agencies would develop an agreement allowing their personnel access to each others data thereby allowing for comparison and assistance. As the WWW develops, one might imagine a situation where people could find out basic information about the authority via the WWW. A researcher, such as myself, might be able to access up to the minute public statistics on the housing authority while sitting in my own livingroom simply by opening my WWW connection. The possibilities that are opened up by computerizing a system are endless.

Appendix A

Data Preparation

In Table 5-1 we see the number of steps that were necessary to transfer the data from the format that the BHA uses to a format that could be used by the class. Table 5-3 shows other changes that were made throughout the transfers in order to prepare the data for analysis.

From Format	To Format
wk1 from the BHA	**dbf** in Excel for editing
dbf in Excel	**Info** in ArcView2
Info in Arc/Info	**oracle** for dataprocessing

Table 5-1 Transfer Formats

In Table 4-1 the tables with the original data that were used in the project, and where they came from, are listed. As is mentioned in Table 5-3, data from these 4 original tables were extracted to create 7 tables for class use. These converted tables were in a relational format, and smaller to facilitate class use. These final 7 tables are listed in Table 5-2. The data dictionaries for these data are listed in the Appendix.

CHEHSMH2	208 rows of employment data for 207 households (One row per employer with only one household having a second employer)
CHEMFMH	Characteristics of each head-of-household that has 'ever' lived in a Mission Hill unit. (One row for each of 1418 households)
CUMOVES	Move-in move-out information for each Mission Hill Unit. There are 2515 moves involving Mission Hill units that are recorded in this table.
CHMOVES	Move-in move-out information for each Household that has ever lived in a Mission Hill Unit. There are 2560 moves recorded in this table.
CUNITSMH	Characteristics of each Mission Hill housing unit, 951 units are described and 549 have current clients.
CFAMILY	Characteristics of each member of each household, 4206 family members (including the head of household) are described.
CINCTOT	Total income and number of income sources for each household. Income for 1440 households is reported.

Table 5-2 The Seven Tables of BHA data used by the Class

There is often a great deal of work that must be done in order to prepare data to be used. Data can come in many formats and it is rare that when given data they are in the format you wish to use it in. There are also issues such as data standardization and privacy that must be addressed. The data we received from the BHA included actual names and social security numbers of residents. This needed to be encoded in some way such that

each resident was matched with a unique identifier, that did not give any other information about the person. The BHA used Social Security Number, SSN, as a household index number that connected other data to the Head of Household, such as identifying all the members of a given family and connecting them to apartment and employment data. Therefore this number needed to be altered in such a way that every time the SSN was used, we could translate it into the correct index number.

Program	*Format changes*	*Data changes*
Excel	Adjust font to fixed width-courier to avoid column width confusion Adjust SSN where last digit is wrapped Adjust field names to omit '.' and the similar characters Rename 'index' to hhindex or uindex since 'index' is an oracle keyword	Adjust city-zip where is should be Boston in HEHSMH2 and HEHSOP2 Add BHASEQ field to allow numbering the original rows
Oracle	substitutes for the SSN numbers Spin off pieces of the multi-valued tables so they provide a cleaner look at each household, sources of income, etc.	Add hhindex and uindex values to 2nd, 3rd, row of household and housing unit tables

Table 5-3 Data Changes

Temporary Tables

If we know that we are going to be doing several queries where we only want to address current clients, we could separate them out into a temporary table to do our queries on. This could be done in a matter of minutes with a command such as the following:

```
CREATE TABLE CURRTEN AS
SELECT CHEMFMH.BHASEQ, HHINDEX, CHEMFMH.BEDR, ELIG, SEX,
       BIRTDATE, AGE, MARISTAT, RACE, NEWRENT, FAMICOUN,
       HOUSNO, STRENAME, APARNO, BUILNO
FROM CHEMFMH, CUNITSMH
WHERE CHEMFMH.HHINDEX = CUNITSMH.CURRCLIE
       AND CUNITSMH.CURRCLIE > 0;
```

We could now do all of our queries regarding current clients only on this new table. What we did here was first we indicated that we wanted to create a new table and call it "CURRTEN". Then we told it which rows to include from the two tables we were combining, in cases where the same column name was used in both tables we indicated

which table it should get its data from. Then we told it the tables that we were interested in "FROM CHEMFMH, CUNITSMH". The statement "WHERE CHEMFMH.HHINDEX = CUNITSMH.CURRCLIE" indicates that it should use the columns HHINDEX & CURRCLIE to line up the rows of the two tables. The last line indicates that we only want current clients.

Lookup Tables

In the discussion of answering the questions I gave a brief explanation of making lookup tables. Here is a more technically complete discussion of what lookup tables are and how they are created. Some of these lookup tables were very easy to create. The BHA gave us lookup tables for most of their fields, however, they also substituted the variables for us in most cases. The idea here, is that if a field only has six possible entries, like martial status: DIVORCED, MARRIED, SEPARATED, SINGLE, UNKNOWN or WIDOWED, the person entering the data would then enter only the code for that race, in this case, D, M, P, S, U or W respectively. When the system is asked for that information, it looks up the code in its lookup table and sends the full word or phrase to the screen. This helps to minimize needed storage space and to keep entries standardized. Examples of the sorts of fields that could be encoded this way include what we received, race, marital status, ethnicity, relationship to head of household, family status and condition of previous housing. We also received several lookup tables that relate to the unit.

However, lookup tables are not always available when they would be helpful. We found the field income source to be fairly standardized, but not completely. By this I mean that in 2376 rows, 56 different income sources were listed. There were however only 25 different categories listed. Several of the 56 were actually misspellings of each other. For example, Unemployment was spelled or abbreviated six different ways. These categories could then be combined into 7 groupings of similar things, for example 'pension' and 'retirement' can be grouped together. These groupings were then further aggregated into four main categories. By creating a lookup table of this structure, which can be seen in Table 5-1, we are able to list each individual with their listed source, and to aggregate everyone into groupings of either employed, on welfare, receiving transfer income, or of an unknown source. The entire table can be found in the Data Dictionary Appendix.

SOURNAME	SOURCE	SOURTYPE	SOURCAT
AFCD	AFCD	WELFARE	WELFARE
AFDC	AFCD	WELFARE	WELFARE
ALIMONY	CHLD SUPPT	TRANSFER	TRANSFER/INS
ASSETS	ASSETS	ASSETS	EARNED
Assets	ASSETS	ASSETS	EARNED
CH SUPPORT	CHLD SUPPT	TRANSFER	TRANSFER/INS

Table 5-1 Income Source Lookup Table

Appendix B

Data Dictionaries

SQL> describe CHEHSMH2;

Name	*Format*	*Type*
BHASEQ	NUMBER(38)	sequence number of original BHA data
HHINDEX	NUMBER(38)	BHA household ID
EMPYRNM	CHAR(36)	Employer Name
EMPYRST1	CHAR(33)	Employer Address (not complete)
EMPYRZIP	CHAR(25)	town Employer is in

BHASEQ	*HHINDEX*	*EMPYRNM*	*EMPYRST1*	*EMPYRZIP*
7	1334	MEDICAL RESOURCES	145 TREMONT STREET	BOSTON, MA
10	1785	BOSTON SCHOOL DEPARTMENT	26 COURT STREET	BOSTON, MA 02108
18	2907	BOSTON UNIVERSITY	80 E CONCORD STREET	BOSTON, MA
25	3211	THE CHILDREN'S HOSPITAL	300 LONGWOOD AVENUE	BOSTON, MA 02115
26	3221	WOMEN'S ED & INDUSTRIAL UNION	356 BOYLSTON STREET	BOSTON, MA 02116
27	3252	AETNA WINDOW CLEANING	352 NEWBURY ST	BOSTON, MA 02115

SQL>describe CINCTOT;

Name	*Format*	*Type*
HHINDEX	NUMBER(38)	BHA household ID
SOURCES	NUMBER	number of income sources
INCTOT	NUMBER	annual total

HHINDEX	*SOURCES*	*INCTOT*
75	1	5538
676	2	27801
692	3	44720
888	1	7149
950	1	7280
1045	1	7536
1334	1	2214
1365	1	6724

Employment lookup table

SOURNAME	**SOURCE**	**SOURTYPE**	**SOURCAT**

AFCD	AFCD	WELFARE	WELFARE
AFDC	AFCD	WELFARE	WELFARE
ALIMONY	CHLD SUPPT	TRANSFER	TRANSFER/INS
SOURNAME	**SOURCE**	**SOURTYPE**	**SOURCAT**
ASSETS	ASSETS	ASSETS	EARNED
Assets	ASSETS	ASSETS	EARNED
CH SUPPORT	CHLD SUPPT	TRANSFER	TRANSFER/INS
CHILD SUPP	CHLD SUPPT	TRANSFER	TRANSFER/INS
CHLD SUPPT	CHLD SUPPT	TRANSFER	TRANSFER/INS
DECEASED	INSURANCE	INSURANCE	TRANSFER/INS
DSS CUSDTY	DSSCUSTDY	TRANSFER	TRANSFER/INS
DSS CUSTDY	DSSCUSTDY	TRANSFER	TRANSFER/INS
EAEDC	EAEDC	UNKNOWN	UNKNOWN
EMPLOYED	GAINFUL EM	EMPLOYED	EARNED
EXP1/31/96	EXP1/31/96	UNKNOWN	UNKNOWN
GAIN EMPL	GAINFUL EM	EMPLOYED	EARNED
GAINFUL EM	GAINFUL EM	EMPLOYED	EARNED
GEN RELIEF	GEN RELIEF	WELFARE	WELFARE
GEN'LRELF	GEN RELIEF	WELFARE	WELFARE
GEN.RELIEF	GEN RELIEF	WELFARE	WELFARE
INTEREST	ASSETS	ASSETS	EARNED
LTD	LTD	UNKNOWN	UNKNOWN
NO SHOW	NO SHOW	UNKNOWN	UNKNOWN
NO INCOME	NO INCOME	NO INCOME	UNKNOWN
NO SHOW	NO SHOW	UNKNOWN	UNKNOWN
NO SHOW 91	NO SHOW	UNKNOWN	UNKNOWN
NO SHOW 92	NO SHOW	UNKNOWN	UNKNOWN
NO SHOW 93	NO SHOW	UNKNOWN	UNKNOWN
NO SHOW 94	NO SHOW	UNKNOWN	UNKNOWN
NO SHOW 95	NO SHOW	UNKNOWN	UNKNOWN
OTHER INC	OTHER INC	UNKNOWN	UNKNOWN
OTHER INCO	OTHER INC	UNKNOWN	UNKNOWN
PART-TIME	PART-TIME	EMPLOYED	EARNED
PENSION	PENSION	INSURANCE	TRANSFER/INS
RES APPL	RES APPL	UNKNOWN	UNKNOWN
RESIDUAL	RESIDUAL	UNKNOWN	UNKNOWN
RETIREMENT	PENSION	INSURANCE	TRANSFER/INS
RETIREMNT	PENSION	INSURANCE	TRANSFER/INS
SELF EMPLY	SELF EMPLY	EMPLOYED	EARNED
SOCSEC	SSI	INSURANCE	TRANSFER/INS
SOPPORT	CHLD SUPPT	TRANSFER	TRANSFER/INS
SSI	SSI	INSURANCE	TRANSFER/INS
SSI SS	SSI	INSURANCE	TRANSFER/INS
SUPPORT CHLD	SUPPT	TRANSFER	TRANSFER/INS
UNEMP COMP	UNEMPLYMNT	TRANSFER	TRANSFER/INS
UNEMPLOYME	UNEMPLYMNT	INSURANCE	TRANSFER/INS
UNEMPLOYMT	UNEMPLYMNT	INSURANCE	TRANSFER/INS
UNEMPLYMNT	UNEMPLYMNT	INSURANCE	TRANSFER/INS
UNEMPLYMT	UNEMPLYMNT	INSURANCE	TRANSFER/INS
UNEMPLYNNT	UNEMPLYMNT	INSURANCE	TRANSFER/INS
USE & OCCP	USE-OCCP	UNKNOWN	UNKNOWN

VDSABLITY	DISABILIY	INSURANCE	TRANSFER/INS
V A DEATH O	INSURANCE	INSURANCE	TRANSFER/INS
VETERANS	DISABILIY	INSURANCE	TRANSFER/INS
WKMNS COMP	WKMNS COMP	INSURANCE	TRANSFER/INS
Y	UNKNOWN	UNKNOWN	UNKNOWN

Chapter Six Bibliography

Alexander, C.J. "Putting the Byte on Canadian Social Welfare Agencies" *Association for Computing Machinery.* 1990.

Beam, Steve. "Purchasing Computer Systems: An Executive's Guide" *Journal of Housing.* Vol.47 Mar/Apr. 1990. pg. 77-95

Boston Housing Authority. "Overview of Urban Revitalization Grant Application Mission Hill Main Development."

Burgess, John S. "Program Management: Modernization Help for LHAs" *JOH* Sept./Oct. 1993.

Cowan, Janet. "NAHRO's New Computer Information Service" *JOH* November/December 1993.

Ferreira, Joseph Jr. "Database Management Tools for Planning" *APA Journal.* Winter 1990 pg. 78-84

Goodrich, Stan. "One PHA's Experience" *Journal of Housing.* March/April 1989. Pg. 65

Hodgkinson, S.L. "The Role of the Corporate IT Function in the Federal IT Organization", pp1-14 (unpublished article).

Mace, Scott. "Data turns into gold for information detectives" *Infoworld,* July 11, 1994.

Manis, Rod; Schaffer, Evan; Jorgensen, Robert. *UNIX Relational Database Management.* "Application Development in the UNIX Environment" \Prentice Hall, Englewood Cliffs, NJ. 1988.

McDonald, Edward. "Effective Automation Implementation: A Plan for H/CD Agencies: *Journal of Housing.* March/April 1992.

_____. "PHA Technological Crossroads" *JOH.* March/April 1993.

_____. H/CD Computer File "System Transition Issues" *Journal of Housing.* Sept./Oct. 1993.

Rajah, LaTonya C. "H/CD Computer Services Review" *Journal of Housing.* March/Apr. 1995.

Rockart, lectures. "Management Information Systems" Spring 1995. Sloan School of Management, MIT.