1998

City of Redlands Public Works Department: Call log database study

Linda Carol Webster

CITY OF REDLANDS PUBLIC WORKS DEPARTMENT

CALL LOG DATABASE STUDY

A Project

Presented to the

Faculty of

California State University,

San Bernardino

In Partial Fulfillment

of the Requirements for the Degree

Master of Public Administration

---

by

Linda Carol Webster

March 1998

CITY OF REDLANDS PUBLIC WORKS DEPARTMENT

CALL LOG DATABASE STUDY

A Project

Presented to the

Faculty of

California State University,

San Bernardino

---

by

Linda Carol Webster

March 1998

Approved by:

[redacted]

Dr. Clifford Young, Department Chair,
Public Administration

3/4/98
Date

[redacted]

Dr. Walt Stewart, Faculty Advisor,
Information and Decision Sciences

[redacted]

Dr. Clifford Young, Second Reader

ABSTRACT

Public managers are looking for effective ways to management, store, and maintain the large quantities of data generated by the growing number of governmental regulations. Today, effective data management requires a computerized database system. Computerized data management has evolved, as all computer technology has. Starting with computerized file systems and advancing to the relational database model.

The simplicity of the design, and the user-friendly frontend systems offered by relational database management systems, make the relational model the standard in today's database design. The relational model was used to design and develop the Call Log database for the City of Redlands Public Works Department. The database will replace the user-unfriendly system now being used by the department.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter I

INTRODUCTION

Today, because of the growing number of governmental regulations, there is a need to store and maintain volumes of data. The increase in regulations has also increased the reporting requirements and need for good, effective decision making of public agencies. Reporting requirements often require data from many sources. Public managers also use these data sources to make decisions. The decisions of public managers affect private citizens and the business community. Because of this, the public manager must have current, complete, and accurate information to make good, effective decisions.

With the increase in storage and maintenance of data, reporting requirements, and effective decision making, there is a need for managed information. Public agencies, like private businesses, have begun to identify and organize data to meet information needs.

With the realization that data is a valuable resource to their agency, the public manager is looking for ways to efficiently manage so that it can be quickly accessed. Today, efficient data management requires the use of a computerized database. "A database is a shared, integrated

computer structure that houses a collection of raw facts" (Rob and Coronel 4), which are of interest to the end user. Data is stored in one location and can be shared by the entire agency.

Chapter II

DATABASE CONCEPTS

Computer File Systems

The first computerized management of data was in the form of data file systems. Files were stored and maintained on the computer for use in different areas in the organization by data processing (DP) specialists. Programs were written to access these files and produce the required reports and information needs for each area.

As the file system grew in an organization there was a greater demand for the DP specialist's programming skills. When a new report format was needed the existing programs were either modified, or a new program written to produce the new information needs. If the structure of a data file or the data characteristics changed, such as adding a field to store new data, or changing a string field to a numeric field, all the programs that accessed that file would then have to be modified to accommodate the new structure (Rob and Coronel 11). Because all programs that access files are subject to change when the file structure or data characteristics change, the file system is said to be both structural and data dependent (Rob and Coronel 13-14).

Because, data files were owned by different areas in an organization and not shared, the computer file system caused the storage of redundant data. Redundant data occurs when the same data is stored in more than one location within the system. Lack of data integrity or data inconsistency is caused by uncontrolled data redundancy. If an employee's phone number appears in more that one file and the phone number changes, the change would need to be made correctly in each file where the employee's phone number is stored. Having to make a change such as this in more than one location leaves room for errors to occur. The problems with the file system and the intensive programming necessary to maintain a file system make using a database system very desirable.

Database Systems

Unlike the file system with many unrelated files stored separately, a database system stores related files in a "single data repository" (Rob and Coronel 17). Thus, representing a change in how the data is accessed, stored, and managed (17). Database Management System (DBMS) software is used to implement and manage the database system. The DBMS interacts with the end user application and the database, performing functions that maintain the integrity and consistency of the data within the system.

Some of these functions include: data dictionary management; data storage management; data transformation and presentation; security management; and backup and recovery management (Rob and Coronel 23). Various database models have been developed, each leading to better data management.

A database model is a collection of logical forms used to represent the data structure and the relationship between the data within the database. There are two categories of database models, the conceptual model and the implementation model. The focus of the conceptual model is on *what* is represented in the database. During the development of the conceptual model the relationship between the data is determined. There are three relationships, one-to-one, one-to-many, and many-to-many. These relationships represent the logical association between data within the database. Unlike the conceptual model the implementation model focuses on *how* the data are represented or on how the data structures are implemented. Implementation models include hierarchical, network, and relational database models. Both the conceptual and implementation models are used to develop the database (Rob and Coronel 27).

The Hierarchical Database Model

North American Rockwell and IBM developed the hierarchical database model in the late 1960's. This was

the first commercial database model used, and it was the basis for further database development (Rob and Coronel 28). The database structure is a collection of records that can be perceived by the user as hierarchical or upside down tree structure as shown in Figure 2.1.

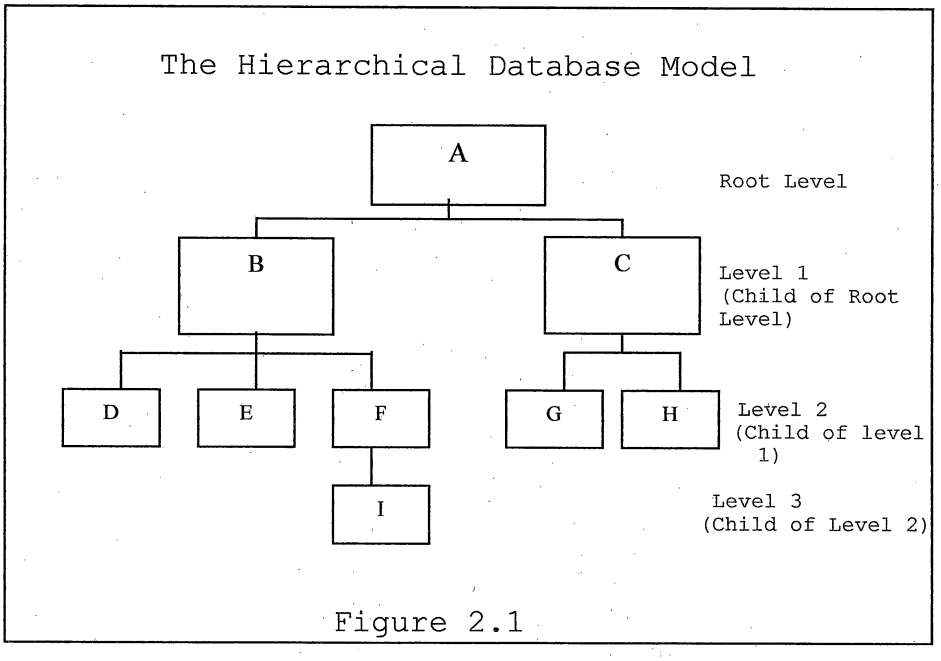


Figure 2.1

The hierarchical database model is said to have a parent-child relationship. Figure 2.1 depicts this relationship. The root level is the parent, lower levels are the children and may also be parents to lower levels. One parent may have many children but a child can only have one parent. Therefore, there can only be one-to-many and one-to-one relationships in this structure.

The advantages the hierarchical database model provided over the file system proved to be many. Because data is stored in a single related repository, data can be shared throughout the organization. With the parent-child relationship there is always a link between parent and child segments of the structure (Rob and Coronel 31). The hierarchical model also decreases the need for programming and program maintenance because data independence can be maintained.

Although the hierarchical database model proved to have many advantages over the file system, it still requires knowledge of the physical structure of the database. If a physical change is made in the structure of the database, changes are also required in all programs that access the database. And because the parent segment must be accessed first, programmers must know the hierarchical path to access data located in a child segment (Rob and Coronel 33). Another shortcoming of the hierarchical model is that many real world relationships do not conform to a one-to-one or one-to-many relationship as required by the hierarchical model. In this model many-to-many relationships are difficult to implement, and may cause restructuring of the database. Also, because of the single-parent rule redundant data must be stored (McFadden and Hoffer 191).

The Network Database Model

The hierarchical database model greatly improved data management over the file system, but it does not effectively represent complex data relationships. The network database model was created to fill this need. The network database model is similar to the hierarchical model, but the network database model violates the single-parent rule (McFadden and Hoffer 197). In the network database model data is represented as sets of record types, each set contains an owner record type and member record types, as shown in Figure 2.2. Each set represents a one-to-many relationship. But unlike the hierarchical database model where a child can only have a single parent, a member may have more than one owner, as shown in Figure 2.3. This in turn reduces the storage of redundant data.

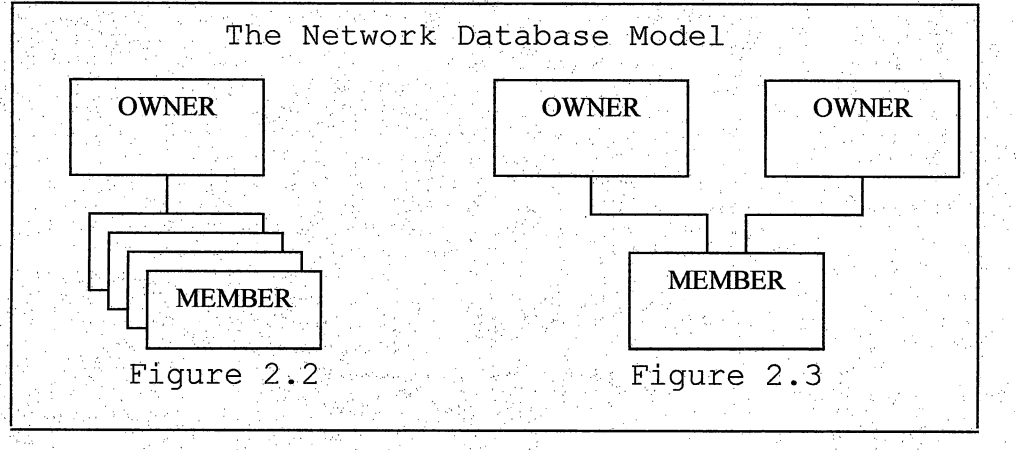


Figure 2.2 Figure 2.3

The network database model has many of the same advantages of the hierarchical database model, but improved some of its defaults. Because a member can have many owners the network database model implements many-to-many relationships more effectively. There is greater data access flexibility. Applications can access an owner record and all its members, if a member belongs to more than one owner it can be accessed through any owner (McFadden and Hoffer 205).

Even though the network database model improves on the hierarchical database model, it too has its downfalls. Although the network database model is data independent, it is structurally dependent. If the physical structure of the database changes, all applications accessing it must also be changed. And because the owner must be accessed before a member can be accessed, programmers must still know the physical structure of the database to create a path for data access. Also, the network database model does not have ad hoc query capabilities. Therefore, if a new data report format is needed, applications would have to be created to meet the new reporting needs. The network database model increased data integrity control and the efficiency of data management, but it was not a user friendly system because of its complexity.

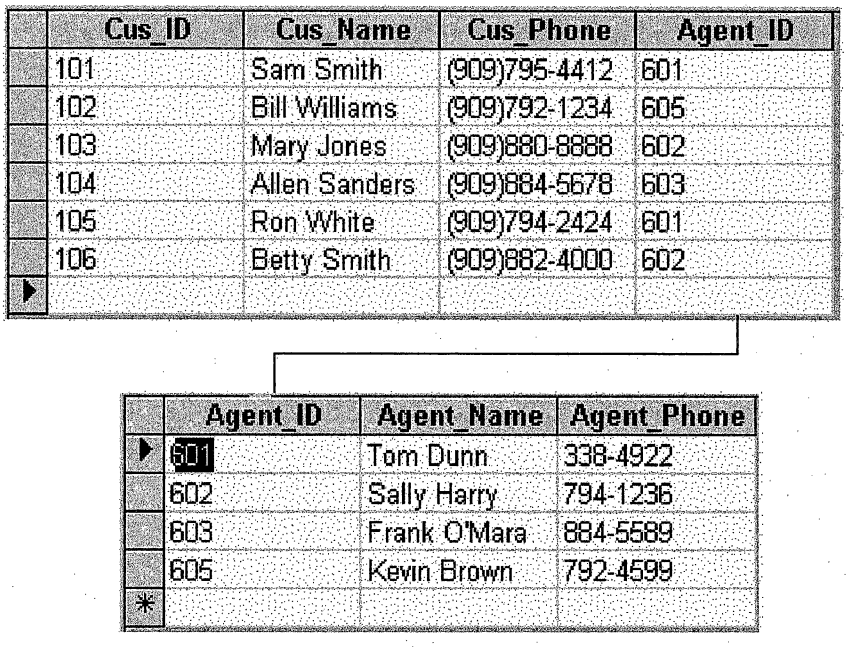
The Relational Database Model

The complexity of the network database model and its lack of ad hoc capabilities lead to the development of the relational database model. The relational database model was developed by E.F.Codd in the 1970's and was a major breakthrough for both users and database designer (Rob and Coronel 38). According to C. J. Date the word to describe the relational database model is "simplicity" (4). The simplicity of the model provides both usability and productivity. Even those with limited computer skills can use the system to manage data and produce required reports. Which in turn increases productivity (Date 5). Because of the simplicity of the design there is no need to know the physical structure of the database. Also, the relational database model has the ability to produce ad hoc queries, limiting the need for extensive programming.

The relational database model eliminates the parent-child and owner-member structures used in the hierarchical and network database models. In the relational database model, the structure is perceived by the user as a collection of related tables containing rows and columns. Each table represents an entity. An entity is a person, place, thing, or event for which data is being stored. Each table stores a collection of related entities. Therefore, a

relational database table resembles a file. Each row within the table represents a single entity record. And each column represents an entity characteristic or attribute. Tables are related by sharing a common attribute, this is shown in Figure 2.4.

Relational Database Tables



Cus_ID	Cus_Name	Cus_Phone	Agent_ID
101	Sam Smith	(909)795-4412	601
102	Bill Williams	(909)792-1234	605
103	Mary Jones	(909)880-8888	602
104	Allen Sanders	(909)884-5678	603
105	Ron White	(909)794-2424	601
106	Betty Smith	(909)882-4000	602

Agent_ID	Agent_Name	Agent_Phone
601	Tom Dunn	338-4922
602	Sally Harry	794-1236
603	Frank O'Mara	884-5589
605	Kevin Brown	792-4599

Figure 2.4

The CUSTOMER table is related to the AGENT table through the Agent_ID number, which is contained in both tables. Although this is storing redundant data it is kept at a minimum, therefore, redundancy is controlled.

Like the hierarchical and network database models, the relational database model is a single data repository. A major difference between the relational and the hierarchical and the network models is that the user does not need to be concerned with the underlying structure of the database. Because of this, the relational database model is both data and structural independent (Date 15-16).

The relational model offers many advantages over the hierarchical and network models. The data and structural independence is important for both the end-user and programmers. Structural independence allows the end-user direct access to the database. If there is a change in the physical structure of the database there is no need for programmers to make corresponding changes in the applications accessing the database (Date 15). Data independence means that, "users and user programs are independent of the logical structure of the database" (Date 16). If there is a change in the logical data structure existing programs accessing the database do not have to be modified or changed.

Relational systems provide "frontend products - ad hoc query subsystems, report writers, business graphics, spreadsheets, application generators" and "natural language

interfaces" (Date 17). These frontend products are easy to use and easy to learn by the end-user.

A prototype of the database can quickly be developed and shown to the "intended users" (Date 18). This allows the end-users to be involved in development of the system. This insures that the final application meets the end-users requirements.

The simplicity of the relational model leads to its disadvantages. Because the relational model hides most of its complexity from the user, it needs more powerful computers to perform all the data management tasks required to maintain the relational database. Also, "the easy-to-use" environment can become a "liability" (Rob and Coronel 43). Because the relational database is structurally independent, the need for proper database design may be overlooked. A poorly designed database can slow down the system and cause the same problems that are related to the file system. Although the relational model has its downfalls, the advantages greatly outweigh the disadvantages. Because of this the relational model is the current standard in database design.

Chapter III

DATABASE DESIGN

The scope and boundaries of this project is to create a database for the City of Redlands Public Works Department. The database design will cover only the daily call log operation. And will be independent of any other database used by the city. The primary users of the database will be the Public Works Departments secretaries.

Mission Statement

> Employees of the City of Redlands are committed to responding to the changing needs of our community, to providing excellent customer service through professionalism, integrity and efficient use of resources, and ensuring Redlands remains a distinctive place in which to live and work.
>
> City of Redlands Mission Statement

Organizational Background

The Public Works Department is one of seven City departments providing service to the citizens of Redlands. The department is comprised of six divisions (see Appendix A, Exhibit 1, Organizational Chart) that are related, yet have specific and separate functions. These divisions include: Engineering Administration, Airport; Building Maintenance, Cemetery, Parks and Street Trees, Streets, and Electrical. Fifty-six full-time and 16 part-time employees

with a variety of professional, technical, clerical, and labor backgrounds staff the department (Mutter).

There are currently two full-time secretaries in the Public Works Department. Their level of computer literacy is low, and their knowledge of computer applications is limited to WordPerfect 6.0 for MS-DOS. In order to help meet the objectives of the cities mission statement, the secretarial staff of the department takes calls from citizens and city employees regarding problems pertaining to:

- Building Maintenance
- Street Trees - Planting and Trimming
- Palm Tree Trimming
- Concrete Maintenance
- Street Light Maintenance
- Street Maintenance
- Street Sweeping

The information is recorded in a Call Log (see Appendix A, Exhibit 2). The Call Log includes: location of problem, this may be a street address or general location; date call received, currently lists numeric month and day; description of problem; caller name and telephone number; and resolved, in which any special remarks are noted by the secretaries or persons receiving the call. Currently the department

receives 10 to 20 calls daily pertaining to problem areas within the city (Johnson).

Current System

The current system used for the Daily Call Log is automated. The information is entered into tables in individual WordPerfect file. Processing is a combination of real-time and batch, depending on who takes the call.

Daily, copies of the Call Log are distributed to the division superintendents specifying calls pertaining to their specific divisions. The superintendents use the Call Log as a guide to determine work assignments within their division. When a problem has been resolved the superintendent marks it off on the Call Log and may or may not indicate what was done and the date completed. The form is then returned to the department secretary. The secretary then enters any remarks made by the superintendents and black lines the complaint, indicating that the problem has been resolved (Johnson). See Appendix A, Exhibit 3 for Data Flow diagrams.

Current Hardware and Software

Hardware	**Software**
2 IBM-compatible stand-alone PC's	MS-DOS 5.0 Operating System
Intel 486™ 66MHz Processors	WordPerfect 6.0 for DOS
400 MB Hard Disk Drives	

16 MB RAM

14 inch VGA Monitors

Mouse

2 HP Laser printers

Microsoft Window 3.11

Microsoft Word 6.0

Microsoft Access 2.0

Current System Problems

There are many problems with the current Call Log system

- Redundant information is maintained, which leads to data inconsistency.
- The system is hard to keep up-to-date because individual logs are maintained for each division.
- It is hard to perform searches. Logs can only be search by phone number and street address. If the site of a problem is a location, such as corner of 6th and State Street, the file must be searched line-by-line. This also must be done if street name is misspelled.
- The system does not allow ad hoc queries.
- Historical data is hard to track because of the individual logs.
- Redundant information is given to division superintendents, because logs are printed by page, not by date.
- System locks up on a regular basis.

Because of the problems with the current system the head of the Public Works Department would like a database designed for the specific purpose of maintaining the Call Log and producing documents pertaining to it.

The use of a relational database in maintaining the Call Log could solve these problems. Historical data could be tracked easily. Searches could be performed on various fields, and a relational database would reduce redundant data being stored and output to divisions. A relational database can produce a variety of reports. It would also provide easy accurate updating of files.

Constraints

Hardware and Software

- The new system must be developed with the existing hardware and software.
- The system is to be a stand-alone system.

Upgrade Capabilities

- The new system must allow conversion to a later version, if necessary.

Information Needs and User Requirements

The general system requirements for the Public Works Call Log database are as follows:

1. The system must be easy to use.
2. The system must have a graphical user interface.

3. The system must be password protected to provide security.
4. The system must reduce redundant data entry and updates.
5. The system must have the ability to produce ad hoc queries.
6. The system must produce the following output
    - Daily Call Logs for building maintenance, concrete maintenance, street light maintenance, street maintenance, street sweeping, palm tree trimming, street tree trimming, and street tree planting.
    - Report by Division
    - Liability Report

The database currently being designed for the Public Works Call Log will be developed using Microsoft Access 2.0 for Windows. This choice has been made because of the constraints of the current system. If the system is upgraded at some time in the future the database can be converted to a newer version of Access.

Call Log Entities and Relationships

An entity is a person, place, thing, or event that data is stored about. The Entity Relationship diagram represents the entities and the relationship between them, one-to-one (1:1), one-to-many (1:N), and many-to-many (N:M). In a

relational database if a many-to-many relationship exits between two entities, redundant data will be stored in both tables. Therefore, to eliminate the storage of redundant data another table is created, called an intersection table, which makes the relationship one-to-many, as seen in Figure 3.1. One complaint can involve many divisions and one division may be involved in many complaints. Thus, a many-to-many relationship exists between the two entities COMPLAINT and DIVISION. To solve this problem an intersection table called INVOLVES was created. The addition of the intersection table not only eliminates the many-to-many relationship, but also, will keep a historic record of the divisions involved in each complaint.

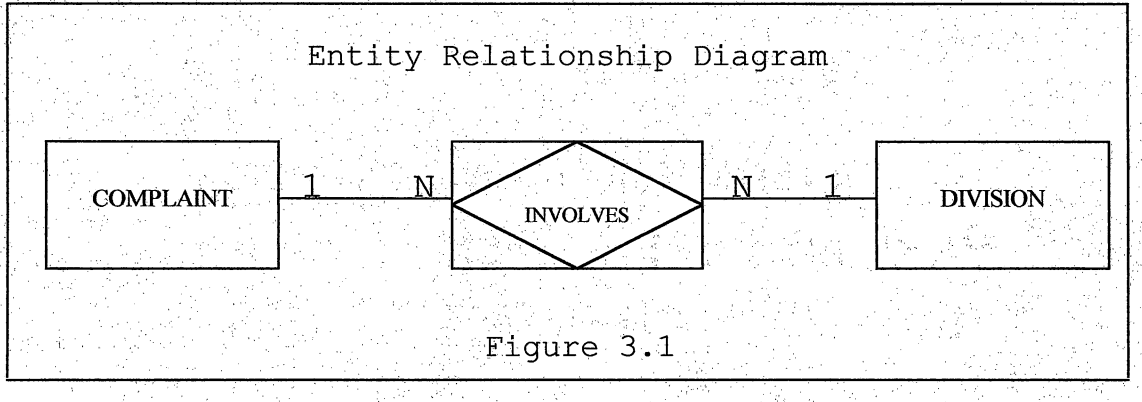


Figure 3.1

Table Definitions

See Data Dictionary for attribute definitions (Appendix B, Exhibit 1).

COMPLAINT| CNum, Location, Problem, Cdate, RecBy, Lname,

Fname, Phone, NumCalls, JobComp, CompDate, Remarks, Claim

The COMPLAINT table will contain data pertaining to specific complaints. To include: Complaint number, location of complaint, problem, call date, call received by, caller first name, caller last name, number of calls, problem resolved, resolved date, remarks, claim (see Appendix B, Exhibit 2).

INVOLVES| CNum, DivName

The INVOLVES table will contain complaint number and division name. This table eliminates the many-to-many relationship between the COMPLAINT and DIVISION tables. It will also contain the historic data on which divisions were involved in specific complaints (see Appendix B, Exhibit 3).

DIVISION| DivName, SuperName

The DIVISION table will contain the names of the division names involved in specific complaints and the name of the division superintendent (see Appendix B, Exhibit 4).

User Views

Because the users of this database have access to the entire database, there is only one user view, depicted in Figure 3.1.

Form Views

The Public Works Call Log database has seven form views.

1. **Main Switchboard** - This form allows access to all other forms in the application. It acts like a main menu in other applications (See Appendix C, Exhibit 1 for user interface). This form also gives the user access to the database window (See Appendix C, Exhibit 8). This will allow the user access to ad hoc query functions and generate new reports.
2. **Daily Call Log form** - This form will be used for data entry of complaint. Data entered into the form will be stored in the COMPLAINT and INVOLVES tables (see Appendix C, Exhibit 2 for user interface).
3. **Update Call Log form** - This form will be used for updating data pertaining to complaints. No new records can be added to the database through this form. Updating of records can also be done in the Daily Call Log form. This form is included in the design for use in the event that the secretary's computers are networked (see Appendix C, Exhibit 3 for user interface).
4. **Division form** - This form will be used for updating the superintendent name. And in the event that complaints are needed to be tracked for a new division, the new division

name and superintendent can be added to the database through this form. Data entered into this form will be stored in the DIVISION table (see Appendix C, Exhibit 4 for user interface).

5. **Print Call Log form** - This form will be used to print the Daily Call Log reports. The user will be prompted for a call date, only records that meet the call date criteria will be printed. The form allows both report preview and print options (see Appendix C, Exhibit 5 for user interface). Queries for the Daily Call Log reports include the following tables and attributes:

   COMPLAINT| CNum,Location, Problem, Cdate, Remarks,
   JobComp = "NO", FName, LName, Phone

   INVOLVES| DivName

   DIVISION| SuperName

   (See Appendix D, Exhibit 1 for report format).

6. **Print Reports form** - This form will be used to print the queried reports, Report by Division and the Liability Report. The user will be prompted for starting and ending dates. Only records that meet the date criteria will be printed. The form allows both report preview and print options (See Appendix C, Exhibit 6 for user interface). Report queries include the following tables and attributes:

Report by Division:

COMPLAINT| CNum, Problem, Cdate, CompDate

INVOLVES| DivName

(See Appendix D, Exhibit 2 for report format).

Liability Report:

COMPLAINT| Location, Problem, Cdate, FName, LName, Phone,

Claim = "YES"

(See Appendix D, Exhibit 3 for report format).

7. **Back Up Dialog Box form** - This form will be used for backup purposes. The form has options for backing up both the entire database or records only (see Appendix C, Exhibit 7 for user interface)

Storage Requirements

The following calculations were used to determine storage requirements. First, the record size was calculated. Each field in the file has a maximum length, that is the number of character allowed. The length of each field is listed below. Each character requires 1 byte of storage space, long integers require 4 bytes and short integers require 2 bytes of storage space. To calculate the record size, sum the length of each field contained in the record. Second, the number of records to be stored was determined. The secretaries in the Public Works Department take 10 to 20 calls a day regarding complaints. To

calculate the number of records, the number of calls were multiplied by the record size. The maximum number of calls, 20, was used to calculate the number of records. Next, the total storage space required per month was calculated. To do this the record length was multiplied by the number of records per month.

To calculate the maximum number of records that can be stored on 3 ½ inch floppy diskette for backup, the total disk capacity (1,440,000 bytes) was divided by the record size.

The following is an estimate of data storage requirements for the Call Log database.

TABLE	ATTRIBUTE	SIZE	STORAGE
COMPLAINT	CNum	Long Integer	4 bytes
	Location	Char(255)	255 bytes
	Problem	Char(255)	255 bytes
	Cdate	Short Integer	2 bytes
	RecBy	Char(2)	2 bytes
	LName	Char(15)	15 bytes
	FName	Char(15)	15 bytes
	Phone	Char(8)	8 bytes
	NumCalls	Char(2)	2 bytes
	JobComp	Char(3)	3 bytes
	CompDate	Short Integer	2 bytes

	Remarks	Char(255)	255 bytes
	Claim	Char(3)	3 bytes
Total record length:			821 Bytes
Estimated number of records per month:			620
Total space required:			509020 bytes or 499 KB
INVOLVES	CNum	Long Integer	4 bytes
	DivName	Char(25)	25 bytes
Total record length:			29 bytes
Estimated number of records per month:			620
Total space required:			17980 bytes or 18 KB
DIVISION	DivName	Char(25)	25 bytes
	SuperName	Char(25)	25 bytes
Total row length:			50 bytes
Number of rows:			9
Total space required:			450 bytes or .44 KB
Total required storage space per month:			527450 bytes or 516 KB
Maximum number of records stored on 3 ½ inch diskette for backup:			1,694 records

It is the department policy to backup the Call Log database once a week. This backup will be used for recovery purposes in case of a system failure. Backup of completed records is done once a month. Records in the COMPLAINT

table where JobComp equals "YES", and records with corresponding CNum in the INVOLVES table will be removed from the database and stored on a 3 ½ inch floppy diskette.

Call Log Database Implementation

As stated earlier, because of system constraints, the Public Works Call Log database was created using Access 2.0, using macros and event procedures to perform routine and repetitive tasks, such as printing Daily Call Log reports (see Appendix E). The database contains three related tables COMPLAINT, INVOLVES, and DIVISION (see Appendix B, Exhibit 1, data dictionary for table attributes and properties). Prototypes of the database were developed and shown to the end-users, resulting in adjustments in the final design.

The final database has been tested and evaluated for data entry and updating; backup and recovery; and security. The database has proven to perform properly in all instances.

Call Log Database Operation

The Call Log database was put into operation on January 16, 1998. The new system is running parallel with the old system. All new calls are entered into the Call Log database. The old system will be maintained until all calls currently stored in it have been resolved.

A user manual was also developed to accompany the Call Log database (see Appendix F). The manual describes the form functions and application instructions, such as steps to backing up the database. The manual also contains the macros and event procedures used in the database.

A three hour training session for the department secretaries was scheduled on the day of installation. The training consisted of a demonstration of the database and hands on training for each secretary. During the hands on portion of the training the secretaries entered data into the various forms, made updates, performed searches, backed up the database, and printed reports.

Chapter IV

CONCLUSION

With the volumes of data necessary to maintain effective reporting and decision making requirements, public managers are looking to computerized data management. The use of a computerized database is the most efficient way to store, maintain, and manage large quantities of data. The use of computer databases also allows data to be shared by the entire agency. Which reduces the need to store redundant data, and preserves computer resources.

Like all other computer technology, computerized management of data has evolved. Starting with the computerized file systems which proved to be inefficient because the unrelated files caused data inconsistency, and the intensive programming necessary to maintain these systems. Database systems, unlike the file system with unrelated files stored separately, stores related data in a single repository. The hierarchical database model offered many advantages over the file system, but it lacked the ability to handle complex data relationships effectively. The network database model was created to fill this need. The network model had the ability to handle more complex data relationships. But provides an unfriendly user work environment. The relational model was considered a major

break through for both users and database designers. Because of the simplicity of the design there is no need to know the physical structure of the database. The relational model offers a user-friendly environment, in which even those with limited computer skills can manage data and produce required reports. Because of this the relational model is considered the standard in database design.

After an analysis of the current system used by the City of Redlands Public Works Department to maintain their call log, which was maintained in separate WordPerfect files, it was decided that they would benefit by a relational database. The relational model will provide the current information needs more efficiently and meet current user requirements. The relational model provides ad hoc query capabilities, which will allow the secretaries to produce reports other than the queried reports available in the database. The relational model provides an easy to use, easy to learn, user-friendly environment. This was necessary because of the limited computer skills of the end-users. Because of current system constraints, the Call Log database was developed in Microsoft Access 2.0. In the event that there is a system upgrade, the database can be converted to later versions of Access.

A P P E N D I X A

ORGANIZATIONAL CHART

CURRENT CALL LOG

DATA FLOW DIAGRAMS

EXHIBIT 1

ORGANIZATIONAL CHART

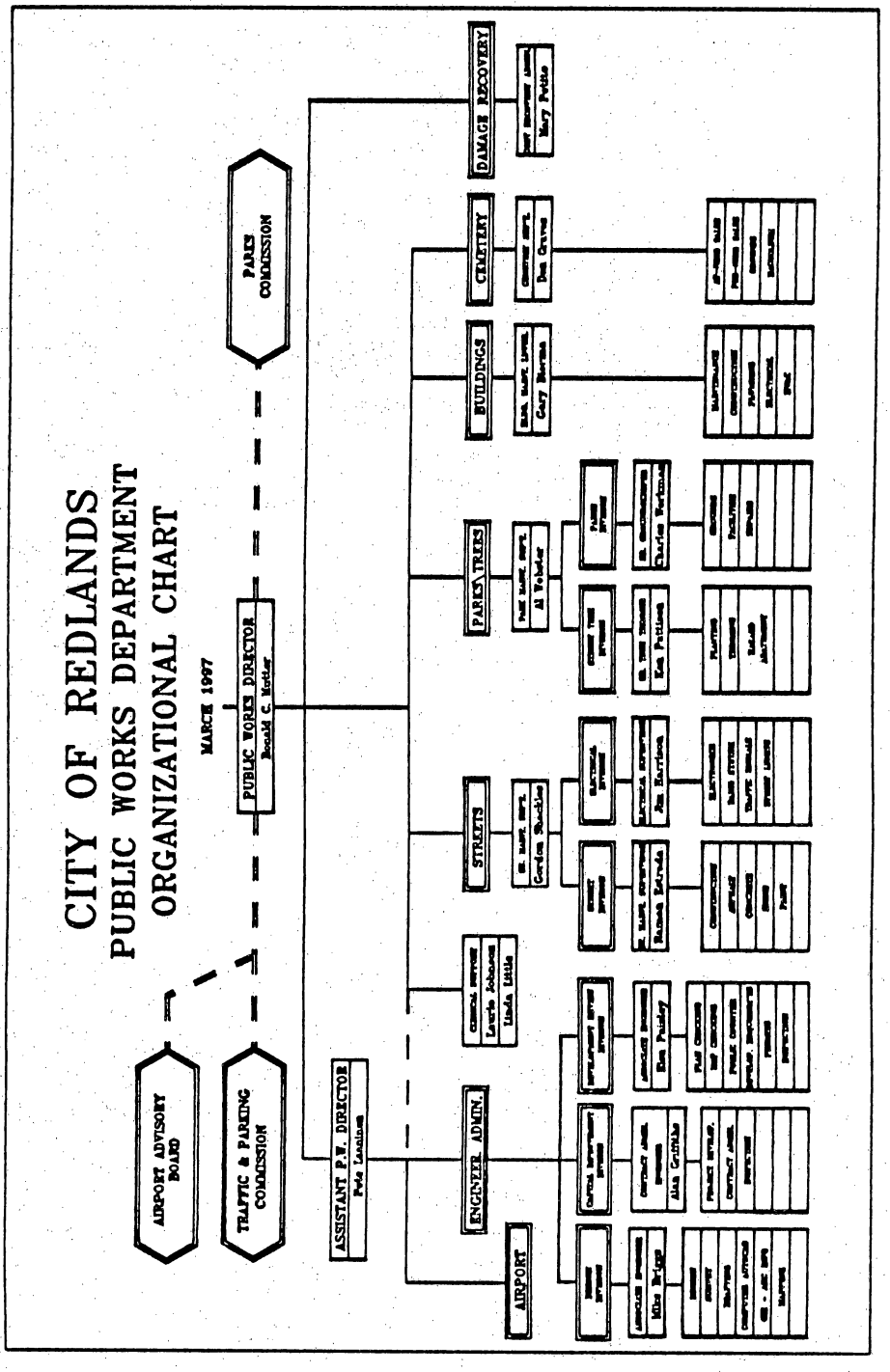


EXHIBIT 2

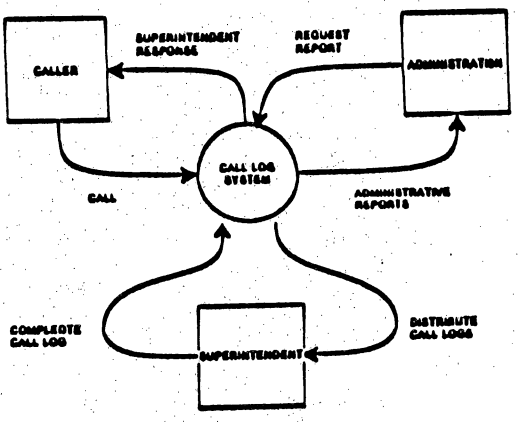
CURRENT DAILY CALL LOG

PALM TREE TRIMMING CALL LOGS FOR 1997

LOCATION	DATE	PROBLEM	NAME/PHONE #	RESOLVED
C/O Monterey & La Cresta Dr.	01.30	Palm Fronds		
Fawn Ct.	01.29	Palm fronds		
510 Fountain Ave.	01.29	Palm fronds		
109 Cajon St.	01.28	Palm trees need to be trimmed. People have slipped and fell because of the large amount of berries on the ground.		
1640 Arbor Dr.	01.28	Palm fronds		
1344 E. Palm Ave.	01.27 01.23	Palm fronds.	[redacted]	Picked up 1-27-97
715 E High	01.22	Palm trees needs to be trimmed.		
1527 Ridge St.	01.22	Palm fronds in empty lot		
411 Brookside Ave.	01.15	Palm fronds in the alley.		
6th & Vine to Cajon	01.14	Palm trimming		
On Cook & Grove behind 1255 East Citrus/Countrywood Apts.	01.02, 12.17	Palm trees need trimmed		
Dearborn from Lisa to Citrus	12.31	Palm trees (not tall) need trimmed - fronds slap people in the face		
1026 Cedar Avenue	12.23	Palm fronds/skirt down at parkway - hit her new van, but she's not sure if it's scratched or dented. I gave her Finance Dept.'s phone number		Radio'd 471

EXHIBIT 3

DATA FLOW DIAGRAMS



CONTEXT DIAGRAM

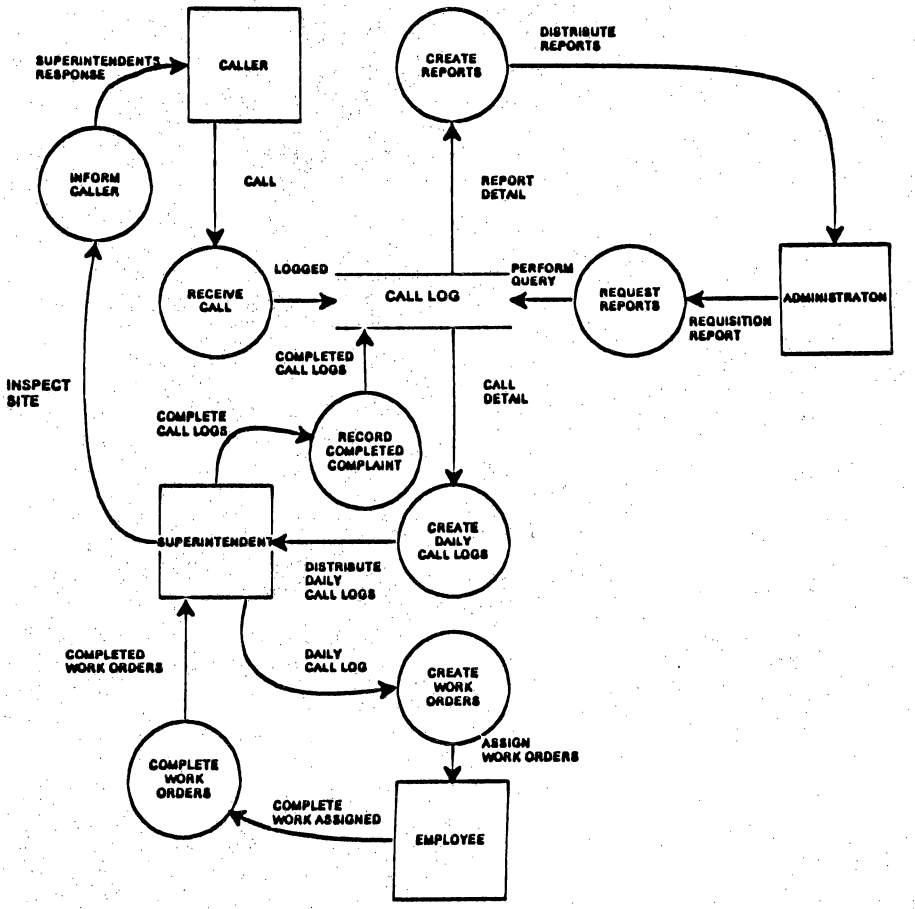


DIAGRAM 0

A P P E N D I X B

DATA DICTIONARY AND TABLE STRUCTURES

EXHIBIT 1

DATA DICTIONARY

ATTRIBUTE	DATA TYPE	SIZE	DEFINITION	INTEGRITY RULES
CNum	Counter		Complaint number Primary Key in the COMPLAINT Table. Concatenated Key INVOLVES table. Caption Complaint Number	Required field, computer will generate.
Location	Text	255 char	Location of complaint. Location may be either an address or description, i. e, 2 W Fern, or third light from corner of Fern and Cajon. COMPLAINT table.	Required field
Problem	Text	255 char	Description of problem. COMPLAINT table.	Required field
Cdate	Date/Time		Date Call receive, computer will generate current date. Caption Date. COMPLAINT table.	Required field MM/DD/YY
RecBy	Text	2 char	Initials of person receiving the call. Caption Received By. COMPLAINT table.	Input Mask: >CC, data entered will automatically be capitalized.
LName	Text	15 char	Last name of caller. Caption Last Name. COMPLAINT TABLE	
FName	Text	15 char	Callers first name. Caption First Name. COMPLAINT table	
Phone	Text	8 char	Caller Phone number. May be a phone number or extension, i.e., 793-3333 orx3333. COMPLAINT table	
NumCalls	Text	2 char	Number of calls received regarding a specific complaint. COMPLAINT table	
JobComp	Text	3 char	Has complaint been resolved, Caption Resolved. COMPLAINT table.	Input Mask >CCC Valid text Yes/No. Default value NO
CompDate	Date/Time		Date complaint resolved. Caption Date Resolved. COMPLAINT table	MM/DD/YY

EXHIBIT 1 DATA DICTIONARY, continued

ATTRIBUTE	DATA TYPE	SIZE	DEFINITION	INTEGRITY RULES
Remarks	Text	255 char	Remarks regarding problem as deemed necessary by person taking call. And any remarks made by superintendent regarding resolution of problem. COMPLAINT table	
Claim	Text	3 char	Probable liability claim. COMPLAINT table	Input Mask >CCC Valid text Yes/ No Default value NO.
DivName	Text	25 char	Name of division involved in complaint. Caption Division Name. Primary Key DIVISION table, Concatenated Key INVOLVES table	Required field Valid text: Concrete Maintenance, Building Maintenance, Street Light Maintenance, Street Maintenance, Street Sweeping, Palm Tree Trimming, Street Tree Trimming, Street Tree Planting.
SuperName	Text	25 char	Name of division superintendent. Caption Superintendent Name. DIVISION table	

EXHIBIT 2

COMPLAINT TABLE

	Complaint Numbe	Location	Problem	Date	Remarks
	18	Treasurer Office	Light out in office anc	7/22/97	
▶	19	2255 W State Stree	Palm frons down bloc	1/14/98	called 408
	20	Redlands Blvd. Cor	Street Light out	11/8/97	
	21	567 W Sterling	Wants to know what	11/9/97	Al gave him a list of
	23	14 East South Ave	Sidewalk raised 2-3"	1/1/98	
	24	200 Delaware	When is the gravel b	1/1/98	called 412
	25	891 E Colton	Tree cut down after s	1/1/98	
	26	233 Anita Ct.	Tree cut down after s	1/1/98	
	27	125 Hastings	Ash tree blew over. '	1/1/98	
	28	1215 Colgate	S/W raised by roots	1/1/98	
	40	9556 W. Palm	Sidwalk raised by tre	1/15/98	Called Interstate
*	(Counter)				

Resolved	Date Resolved	Number of Calls	First Name	Last Name	Phone
YES	1/14/98	1	Peggie		x7657
NO		1	Sam	Hill	792-0999
NO		1	Jim		x3457
YES	1/14/98	1	Tom	Sentmen	793-8765
NO		1	Bob	Engel	335-0202
NO		1	Sam	Collins	794-8989
NO		1	Pat	Darling	793-2277
NO		1	Diana	Jones	792-4765
NO		1	Sally	Ennis	792-3308
NO		1	Shirlene	Mobley	792-0585
NO		1	Tom	Miller	793-3333
NO		1			

Received By	Claim
LW	NO
LW	NO
LW	NO
LW	NO
LW	NO
LW	YES
LW	NO
LW	NO
LW	NO
LW	NO
LW	NO
	NO

EXHIBIT 3

INVOLVES TABLE

	CNum	Division Name
▶	18	Building Maintenance
	19	Palm Tree Trimming
	20	Street Light Maintenance
	21	Street Tree Planting
	23	Concrete Maintenance
	23	Street Tree Trimming
	24	Concrete Maintenance
	25	Street Tree Planting
	26	Street Tree Planting
	27	Street Tree Planting
	28	Concrete Maintenance
	28	Street Tree Trimming
	40	Concrete Maintenance
	40	Street Sweeping
	40	Street Tree Trimming
*	0	

EXHIBIT 4

DIVISION TABLE

	Division Name	Superintendent
	Building Maintenance	Gary Banks
	Concrete Maintenance	Tom Smith
	Palm Tree Trimming	Al Weber
	Street Light Maintenance	Tom Smith
	Street Maintenance	Tom Smith
	Street Sweeping	Tom Smith
	Street Tree Planting	Al Weber
	Street Tree Trimming	Al Weber
*		

A P P E N D I X C

USER INTERFACES

EXHIBIT 1

MAIN SWITCHBOARD

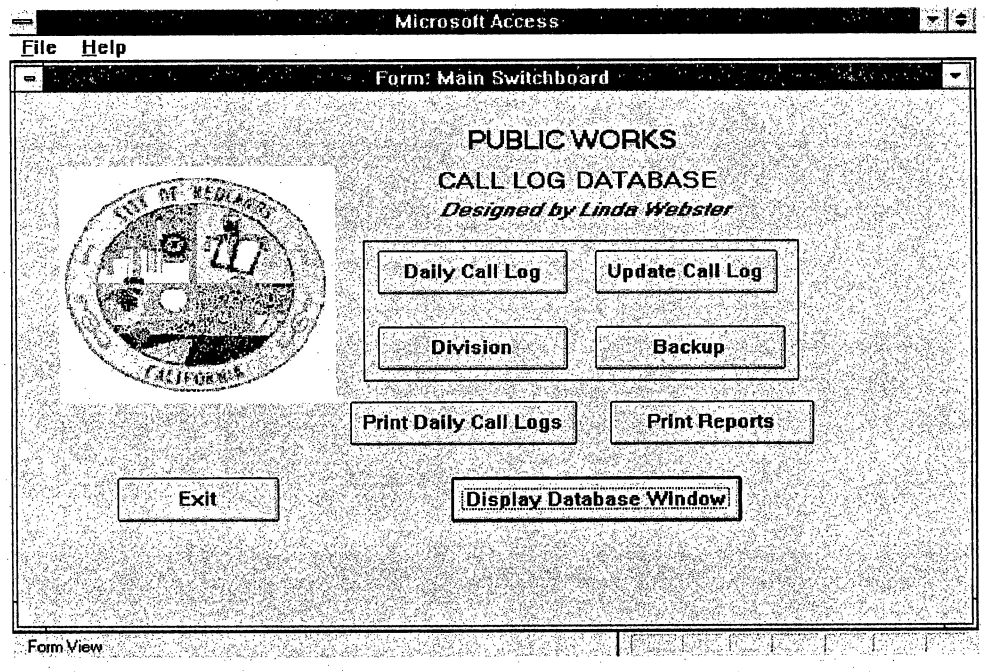


EXHIBIT 2

DAILY CALL LOG FORM

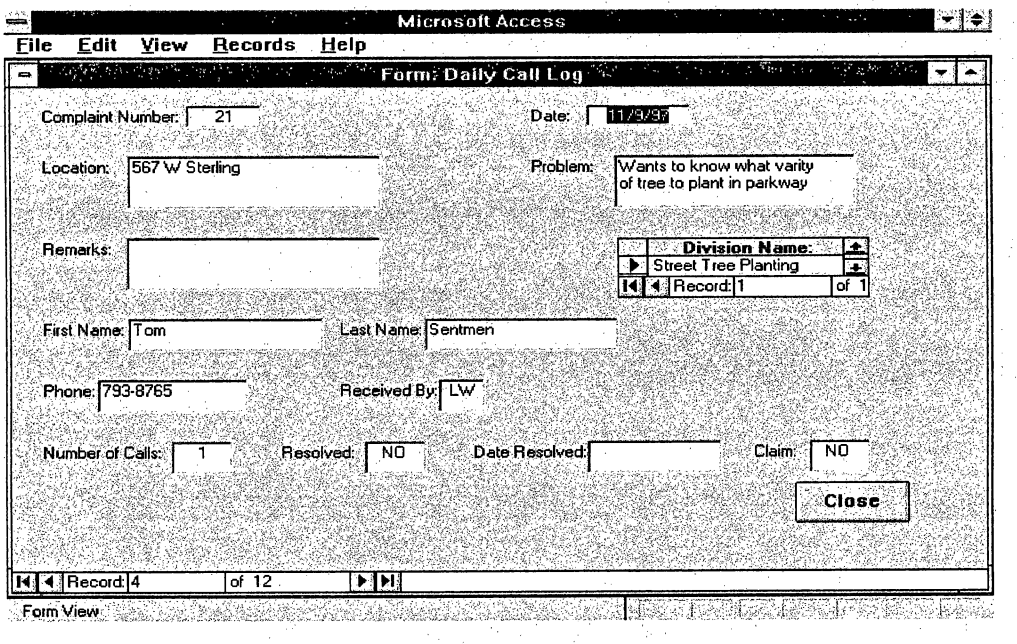


EXHIBIT 3

UPDATE CALL LOG FORM

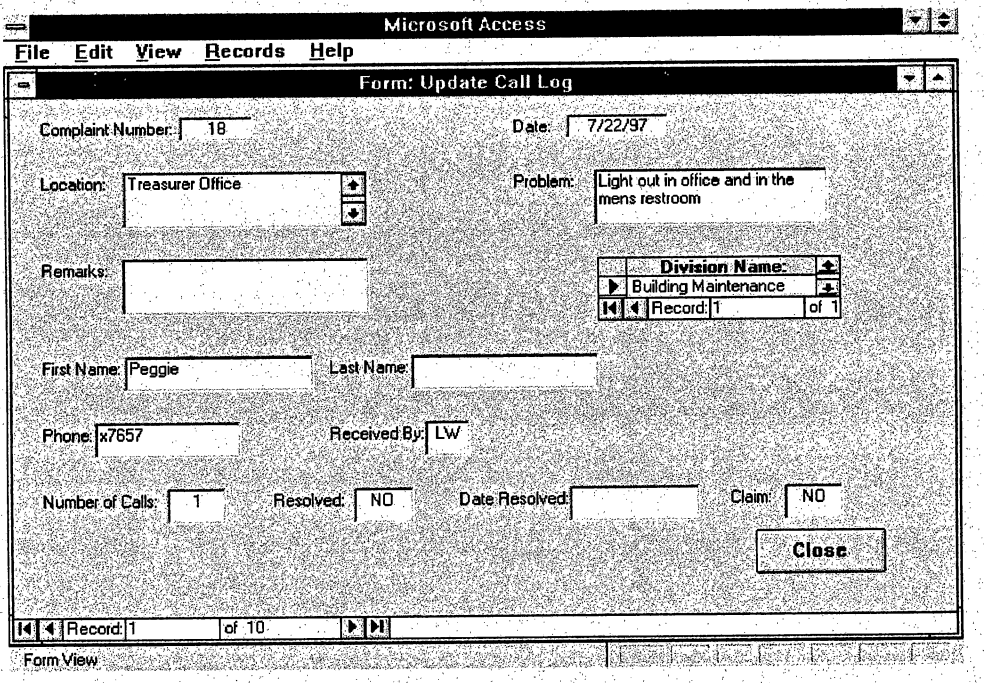


EXHIBIT 4

DIVISION FORM

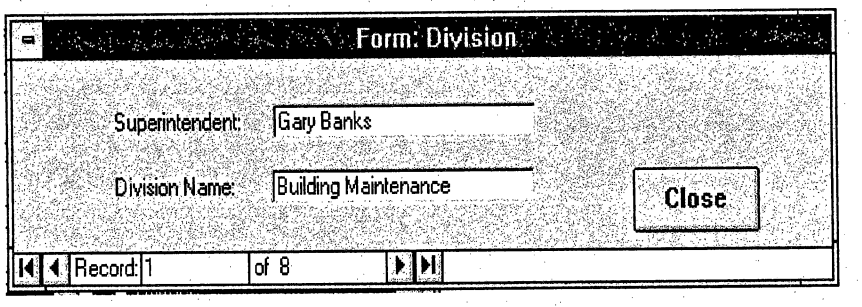


EXHIBIT 5

PRINT CALL LOG FORM

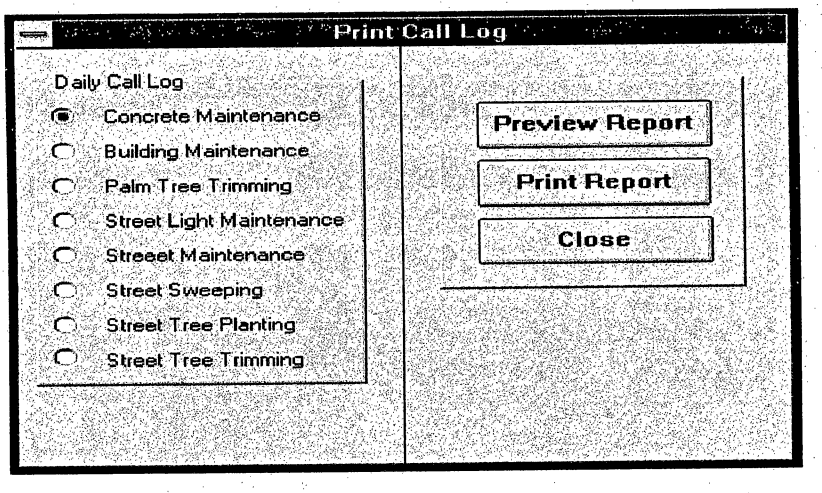


EXHIBIT 6

PRINT REPORTS FORM

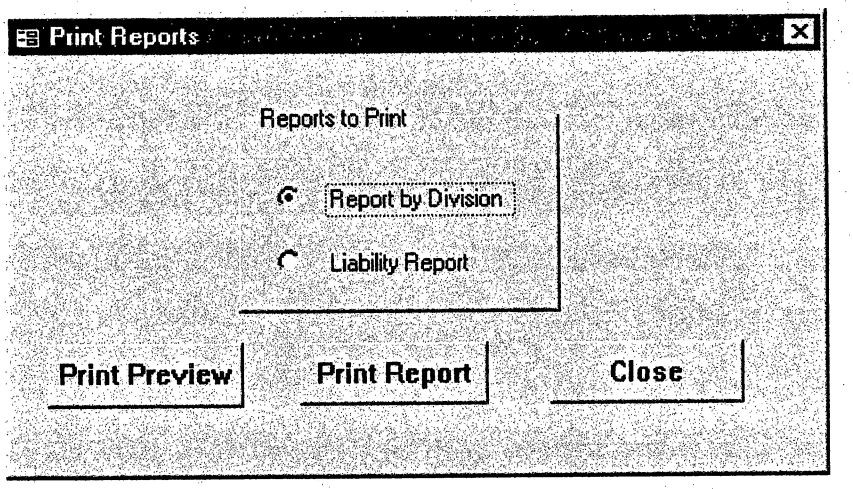


EXHIBIT 7

BACK UP DIALOG BOX

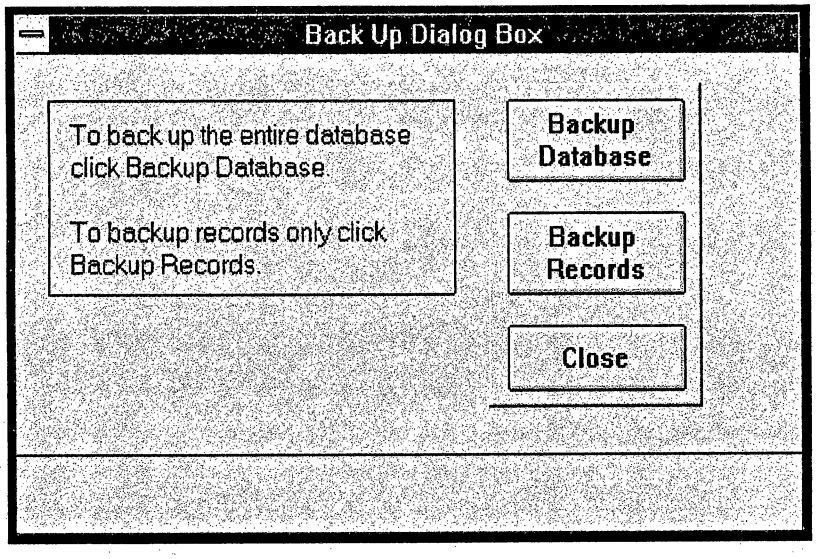
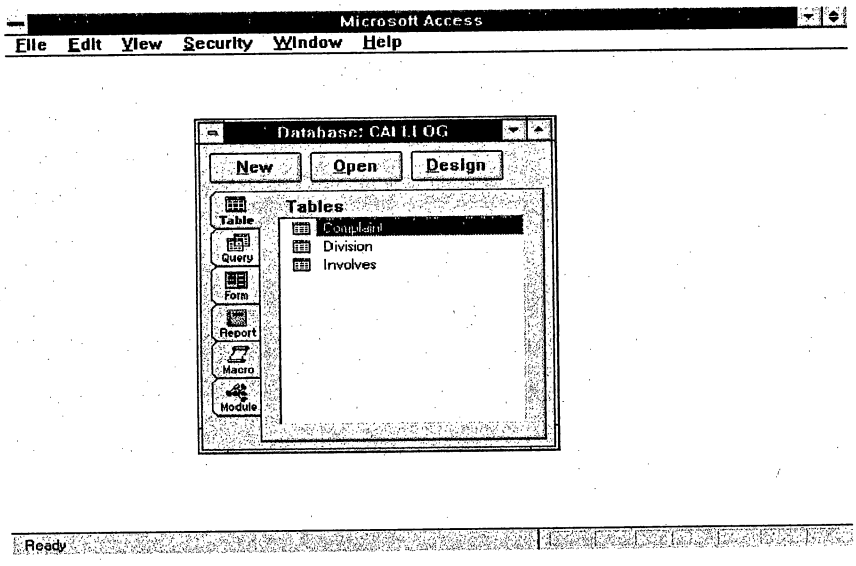


EXHIBIT 8

DATABASE WINDOW



A P P E N D I X D

REPORT FORMATS

EXHIBIT 1

DAILY CALL LOG REPORT

Street Tree Planting Call Log

1/19/98

Al Weber

Comp Num	Location	Date	Name/Phone #	Problem	Remarks
25	891 E Colton	1/1/98	Pat Darling 793-2277	Tree cut down after storm. Wantst Crepe Myrtle planted	
26	233 Anita Ct.	1/1/98	Diana Jones 792-4765	Tree cut down after storm. Wants an Oleander tree.	
27	125 Hastings	1/1/98	Sally Ennis 792-3308	Ash tree blew over. Wants Crepe Myrtle planted	

Report by Division

1/19/98

Concrete Maintenance	Problem	Call Date	Date Resolved
	S/W raised by roots	1/1/98	
	When is the gravel between the curb & pavement going to be compleded on Delaware? Wife fell in gravel area and is going to the doctors today.	1/1/98	
	Sidewalk raised 2-3", sewer affected by tree roots, street lifting and toxic cesspool in street.	1/1/98	

Street Light Maintenance	Problem	Call Date	Date Resolved
	Street Light out	11/8/97	

Street Tree Planting	Problem	Call Date	Date Resolved
	Ash tree blew over. Wnads Crepe Myrtle planted	1/1/98	
	Tree cut down after storm. Wandt an Oleander tree.	1/1/98	
	Tree cut down after storm. Wandt Crepe Myrtle planted	1/1/98	
	Wants to know what varity of tree to plant in parkway	11/9/97	1/14/98

Street Tree Trimming	Problem	Call Date	Date Resolved
	S/W raised by roots	1/1/98	
	Sidewalk raised 2-3", sewer affected by tree roots, street lifting and toxic cesspool in street.	1/1/98	

EXHIBIT 2

REPORT BY DIVISION

EXHIBIT 3

LIABILITY REPORT

Liability Report

1/19/98

Date	Location	Problem	Name/Phone #
1/1/98	200 Delaware	When is the gravel between the curb & pavement going to be compleded on Delaware? Wife fell in gravel area and is going to the doctors today.	Sam Collins 794-8989
1/19/98	2 West State Street	Tree lime fell on customers car.	Bill Morgan 793-2255
1/19/98	234 Fern Ave	Tree roots in sewer lines	Sam Miller 793-9999
1/19/98	562 La Marido	S/W raised by tree roots	May Thompson 792-7654

A P P E N D I X E

MACROS AND EVENT PROCEDURES

Properties

Date Created:	1/4/98 7:25:33 PM	Last Updated:	1/4/98 7:25:33 PM
Owner:	admin		

Actions

Name	Condition	Action	Argument	Value
Autoexec				
	Runs automatically when database is opened			
		RunCommand	Command:	2
	Hide database window			
		OpenForm	Form Name:	Main Switchboard
			View:	Form
			Filter Name:	
			Where Condition:	
			Data Mode:	Edit
			Window Mode:	Normal
	Opens Main Switchboard form			

User Permissions

admin

Group Permissions

Admins
Users

Properties

Date Created:	12/31/97 4:59:42 PM	Last Updated:	1/11/98 5:13:25 PM
Owner:	admin		

Actions

Name	Condition	Action	Argument	Value
Database Window				
		SetValue	Item:	[Visible]
			Expression:	No
	Hide Main Switchboard			
		SelectObject	Object Type:	Form
			Object Name:	
			In Database Window:	Yes
	Display Database Window			

User Permissions

admin

Group Permissions

Admins
Users

Properties

Date Created:	12/28/97 2:48:38 PM	Last Updated:	1/8/98 12:55:10 PM
Owner:	admin		

Actions

Name	Condition	Action	Argument	Value
	Attach to Print Button on Print Call Log Dialog Box			
Print		SetValue	Item: Expression:	[Visible] No
	Hide Print Call Log Dialog Box.			
	[Daily Call Log]=1	OpenReport	Report Name: View: Filter Name: Where Condition:	Building Maintenance Print
	Print Building Maintenance Call Log			
	[Daily Call Log]=2	OpenReport	Report Name: View: Filter Name: Where Condition:	Concrete Maintenance Print
	Print Concrete Maintenance CallLog			
	[Daily Call Log]=3	OpenReport	Report Name: View: Filter Name: Where Condition:	Palm Tree Timming Print
	Print Palm Tree Trimming Call Log			
	[Daily Call Log]=4	OpenReport	Report Name: View: Filter Name: Where Condition:	Street Light Maintenance Print
	Print Street Light Maintenance Call Log			
	[Daily Call Log]=5	OpenReport	Report Name: View: Filter Name: Where Condition:	Street Maintenance Print
	Print Street Maintenance Call Log			
	[Daily Call Log]=6	OpenReport	Report Name: View: Filter Name: Where Condition:	Street Sweeping Print

Print Street Sweeping Call Log

[Daily Call Log]=7	OpenReport	Report Name: View: Filter Name: Where Condition:	Street Tree Planting Print

Print Street Tree Planting Call Log

[Daily Call Log]=8	OpenReport	Report Name: View: Filter Name: Where Condition:	Street Tree Trimming Print

Print Street Tree Trimming Call Log

	Close	Object Type: Object Name: Save:	Form Print Call Log

Close Print Reports Dialog Form

Attach to Print Preview Button on Print Call Log Dialog Box

Print Preview

	SetValue	Item: Expression:	[Visible] No

Hide Print Call Log Dialog Box.

[Daily Call Log]=1	OpenReport	Report Name: View: Filter Name: Where Condition:	Building Maintenance Print Preview

Preview Building Maintenace Call Log

[Daily Call Log]=2	OpenReport	Report Name: View: Filter Name: Where Condition:	Concrete Maintenance Print Preview

Preview Concrete Maintenance Call Log

[Daily Call Log]=3	OpenReport	Report Name: View: Filter Name: Where Condition:	Palm Tree Timming Print Preview

Preview Palm Tree Trimming Call Log

[Daily Call Log]=4	OpenReport	Report Name: View: Filter Name: Where Condition:	Street Light Maintenance Print Preview

Preview Street Light Maintenance Call Log

[Daily Call Log]=5	OpenReport	Report Name: View:	Street Maintenance Print Preview

		Filter Name:	
		Where Condition:	
Preview Street Maintenance Call Log			
[Daily Call Log]=6	OpenReport	Report Name:	Street Sweeping
		View:	Print Preview
		Filter Name:	
		Where Condition:	
Preview Street Sweeping Call Log			
[Daily Call Log]=7	OpenReport	Report Name:	Street Tree Planting
		View:	Print Preview
		Filter Name:	
		Where Condition:	
Preview Street Tree Planting Call Log			
[Daily Call Log]=8	OpenReport	Report Name:	Street Tree Trimming
		View:	Print Preview
		Filter Name:	
		Where Condition:	
Preview Street Tree Timming Call Log			
	Close	Object Type:	Form
		Object Name:	Print Call Log
		Save:	
Close Print Call Log Dialog Form			

User Permissions

admin

Group Permissions

Admins
Users

Properties

Date Created:	12/31/97 7:22:08 PM	Last Updated:	1/1/98 6:24:47 PM
Owner:	admin		

Actions

Name	Condition	Action	Argument	Value
	Attach to Print Button on Print Reports Form			
Print Report				
		SetValue	Item: Expression:	[Visible] No
	Hides Print Reports Form			
	[Reports to Print]=1	OpenReport	Report Name: View: Filter Name: Where Condition:	Report by Division Print
	Print Report by Division			
	[Reports to Print]=2	OpenReport	Report Name: View: Filter Name: Where Condition:	Liability Report Print Liability
	Print Liability Report			
		Close	Object Type: Object Name: Save:	Form Print Reports
	Close Print Reports Form			
	Attach to Print Preview Button on Print Report Dialog Form			
Print Preview				
		SetValue	Item: Expression:	[Visible] No
	Hide Print Reports From			
	[Reports to Print]=1	OpenReport	Report Name: View: Filter Name: Where Condition:	Report by Division Print Preview
	Preview Report by Division			
	[Reports to Print]=2	OpenReport	Report Name: View: Filter Name:	Liability Report Print Preview Liability

	Where Condition:	
Preview Liability Report		
Close	Object Type:	Form
	Object Name:	Print Reports
	Save:	
Close Print Reports Form		

User Permissions

admin

Group Permissions

Admins
Users

Properties

Date Created:	1/3/98 7:09:37 PM	Last Updated:	1/11/98 5:13:00 PM
Owner:	admin		

Actions

Name	Condition	Action	Argument	Value
Backup Complaint				
	Attached to Backup Records button on Backup Dialog Box			
		MsgBox	Message:	Please insert a disk into the A drive
			Beep:	Yes
			Type:	None
			Title:	Backup Records
	Displays message box			
		OpenQuery	Query Name:	Append
			View:	Datasheet
			Data Mode:	Edit
	Opens Append query, appends records to Old Complaints			
		OpenQuery	Query Name:	Delete
			View:	Datasheet
			Data Mode:	Edit
	Opens Delete query, deletes old records form Complaint and Involves tables			
		MsgBox	Message:	Records have successfully been copied to the A drive.
			Beep:	Yes
			Type:	None
			Title:	
	Displays message box			

User Permissions

admin

Group Permissions

Admins
Users

Properties

Date Created:	1/4/98 8:04:38 PM	Last Updated:	1/11/98 7:26:01 AM
Owner:	admin		

Actions

Name	Condition	Action	Argument	Value
Backup Database				
	Backup entire database to A:BKUPDB			
		MsgBox	Message:	Please Insert a disk into the A drive
			Beep:	Yes
			Type:	None
			Title:	Backup Database
	Displays message box			
		OpenQuery	Query Name:	BKUP Complaint TBL
			View:	Datasheet
			Data Mode:	Edit
	Backups Complaint table			
		OpenQuery	Query Name:	BKUP Involves TBL
			View:	Datasheet
			Data Mode:	Edit
	Backups Involves table			
		OpenQuery	Query Name:	BKUP Division TBL
			View:	Datasheet
			Data Mode:	Edit
	Backups Division table			
		MsgBox	Message:	Backup is complete
			Beep:	Yes
			Type:	None
			Title:	Backup complete
	Displays message box			

User Permissions

admin

Group Permissions

Admins

Users

Properties

Date Created:	1/8/98 4:07:26 PM	Last Updated:	1/11/98 4:40:47 PM
Owner:	admin		

Actions

Name	Condition	Action	Argument	Value
Division Menu				
	Displays menu items for Update Super form			
		AddMenu	Menu Name: Menu Macro Name: Status Bar Text:	&File Division Menu_File
	Displays File menu items			
		AddMenu	Menu Name: Menu Macro Name: Status Bar Text:	&Edit Division Menu_Edit
	Desplays Edit menu items			
		AddMenu	Menu Name: Menu Macro Name: Status Bar Text:	&Help Division Menu_Help
	Desplays Help menu			

User Permissions

admin

Group Permissions

Admins
Users

Properties

Date Created: 1/8/98 4:07:27 PM Last Updated: 1/8/98 4:07:28 PM
Owner: admin

Actions

Name	Condition	Action	Argument	Value
&Undo		RunCommand	Command:	292
Undo Current Record		RunCommand	Command:	292
-				
D&elete		RunCommand	Command:	337
-				
&Find...		RunCommand	Command:	30
&Replace...		RunCommand	Command:	29
-				
-				
-				

User Permissions

admin

Group Permissions

Admins
Users

Properties

Date Created:	1/8/98 4:07:26 PM	Last Updated:	1/11/98 4:42:11 PM
Owner:	admin		

Actions

Name	Condition	Action	Argument	Value
Division Menu_File				
E&xit		RunCommand	Command:	3

User Permissions

admin

Group Permissions

Admins
Users

Properties

Date Created:	1/8/98 4:07:28 PM	Last Updated:	1/8/98 4:07:29 PM
Owner:	admin		

Actions

Name	Condition	Action	Argument	Value
&Contents		RunCommand	Command:	235
&Search...		RunCommand	Command:	100
C&ue Cards		RunCommand	Command:	235
-				
&Technical Support		RunCommand	Command:	235
&About Microsoft Access...		RunCommand	Command:	35

User Permissions

admin

Group Permissions

Admins
Users

Properties

Date Created:	1/8/98 4:13:35 PM	Last Updated:	1/11/98 4:43:50 PM
Owner:	admin		

Actions

Name	Condition	Action	Argument	Value
MSMenu				
	Displays Main Switchboard Menu Items			
		AddMenu	Menu Name: Menu Macro Name: Status Bar Text:	&File MSMenu_File
	Desplays File menu items			
		AddMenu	Menu Name: Menu Macro Name: Status Bar Text:	&Help MSMenu_Help
	Displays Help menu			

User Permissions

admin

Group Permissions

Admins
Users

Properties

Date Created:	1/8/98 4:13:36 PM	Last Updated:	1/11/98 4:44:18 PM
Owner:	admin		

Actions

Name	Condition	Action	Argument	Value
MSMenu_File				
E&xit		RunCommand	Command:	3

User Permissions

admin

Group Permissions

Admins
Users

Properties

Date Created:	1/8/98 4:13:37 PM	Last Updated:	1/8/98 4:13:37 PM
Owner:	admin		

Actions

Name	Condition	Action	Argument	Value
&Contents		RunCommand	Command:	235
&Search...		RunCommand	Command:	100
C&ue Cards		RunCommand	Command:	235
-				
&Technical Support		RunCommand	Command:	235
&About Microsoft Access...		RunCommand	Command:	35

User Permissions

admin

Group Permissions

Admins
Users

Properties

Date Created:	1/8/98 4:02:52 PM	Last Updated:	1/11/98 4:47:01 PM
Owner:	admin		

Actions

Name	Condition	Action	Argument	Value
New Menubar				
	Displays menu items on Call Log Data Entry form			
		AddMenu	Menu Name: Menu Macro Name: Status Bar Text:	&File New Menubar_File
	Displays File menu items			
		AddMenu	Menu Name: Menu Macro Name: Status Bar Text:	&Edit New Menubar_Edit
	Displays Edit menu items			
		AddMenu	Menu Name: Menu Macro Name: Status Bar Text:	&Records New Menubar_Records
	Displays Records menu items			
		AddMenu	Menu Name: Menu Macro Name: Status Bar Text:	&Help New Menubar_Help
	Displays Help menu			

User Permissions

admin

Group Permissions

Admins
Users

Properties

Date Created:	1/8/98 4:02:53 PM	Last Updated:	1/8/98 4:02:54 PM
Owner:	admin		

Actions

Name	Condition	Action	Argument	Value
&Undo		RunCommand	Command:	292
Undo Current Record		RunCommand	Command:	292
-				
D&elete		RunCommand	Command:	337
-				
&Find...		RunCommand	Command:	30
&Replace...		RunCommand	Command:	29
-				

User Permissions

admin

Group Permissions

Admins
Users

Properties

Date Created:	1/8/98 4:02:53 PM	Last Updated:	1/8/98 4:02:53 PM
Owner:	admin		

Actions

Name	Condition	Action	Argument	Value
-				
E&xit		RunCommand	Command:	3

User Permissions

admin

Group Permissions

Admins
Users

Properties

Date Created: 1/8/98 4:02:56 PM Last Updated: 1/8/98 4:02:56 PM
Owner: admin

Actions

Name	Condition	Action	Argument	Value
&Contents		RunCommand	Command:	235
&Search...		RunCommand	Command:	100
C&ue Cards		RunCommand	Command:	235
-				
&Technical Support		RunCommand	Command:	235
&About Microsoft Access...		RunCommand	Command:	35

User Permissions

admin

Group Permissions

Admins
Users

Properties

Date Created:	1/8/98 4:02:54 PM	Last Updated:	1/8/98 4:02:55 PM
Owner:	admin		

Actions

Name	Condition	Action	Argument	Value
&Data Entry		RunCommand	Command:	78
&Go To		AddMenu	Menu Name: Menu Macro Name: Status Bar Text:	&Go To New Menubar_Records_Go To
&Refresh		RunCommand	Command:	18
-				
-				

User Permissions

admin

Group Permissions

Admins
Users

Properties

Date Created: 1/8/98 4:02:55 PM Last Updated: 1/8/98 4:02:55 PM
Owner: admin

Actions

Name	Condition	Action	Argument	Value
&First		RunCommand	Command:	67
&Last		RunCommand	Command:	68
&Next		RunCommand	Command:	65
&Previous		RunCommand	Command:	66
Ne&w		RunCommand	Command:	28

User Permissions

admin

Group Permissions

Admins
Users

Properties

Date Created:	1/8/98 4:04:59 PM	Last Updated:	1/11/98 4:48:54 PM
Owner:	admin		

Actions

Name	Condition	Action	Argument	Value
Update				
	Displays Update Call Log menu bar			
		AddMenu	Menu Name: Menu Macro Name: Status Bar Text:	&File Update_File
	Displays File menu items			
		AddMenu	Menu Name: Menu Macro Name: Status Bar Text:	&Edit Update_Edit
	Displays Edit menu items			
		AddMenu	Menu Name: Menu Macro Name: Status Bar Text:	&Records Update_Records
	Displays Record menu items			
		AddMenu	Menu Name: Menu Macro Name: Status Bar Text:	&Help Update_Help
	Displays Help menu			

User Permissions

admin

Group Permissions

Admins
Users

Properties

Date Created:	1/8/98 4:04:59 PM	Last Updated:	1/8/98 4:05:00 PM
Owner:	admin		

Actions

Name	Condition	Action	Argument	Value
-				
E&xit		RunCommand	Command:	3

User Permissions

admin

Group Permissions

Admins
Users

Properties

Date Created: 1/8/98 4:05:03 PM Last Updated: 1/8/98 4:05:03 PM
Owner: admin

Actions

Name	Condition	Action	Argument	Value
&Contents		RunCommand	Command:	235
&Search...		RunCommand	Command:	100
C&ue Cards		RunCommand	Command:	235
-				
&Technical Support		RunCommand	Command:	235
&About Microsoft Access...		RunCommand	Command:	35

User Permissions

admin

Group Permissions

Admins
Users

Properties

Date Created: 1/8/98 4:05:00 PM Last Updated: 1/8/98 4:05:00 PM
Owner: admin

Actions

Name	Condition	Action	Argument	Value
&Undo		RunCommand	Command:	292
Undo Current Record		RunCommand	Command:	292
-				
D&elete		RunCommand	Command:	337
-				
&Find...		RunCommand	Command:	30
&Replace...		RunCommand	Command:	29
-				

User Permissions

admin

Group Permissions

Admins
Users

Properties

Date Created:	1/8/98 4:05:01 PM	Last Updated:	1/8/98 4:05:01 PM
Owner:	admin		

Actions

Name	Condition	Action	Argument	Value
&Go To		AddMenu	Menu Name: Menu Macro Name: Status Bar Text:	&Go To Update_Records_Go To
&Refresh		RunCommand	Command:	18

User Permissions

admin

Group Permissions

Admins
Users

Properties

Date Created: 1/8/98 4:05:02 PM Last Updated: 1/8/98 4:05:02 PM
Owner: admin

Actions

Name	Condition	Action	Argument	Value
&First		RunCommand	Command:	67
&Last		RunCommand	Command:	68
&Next		RunCommand	Command:	65
&Previous		RunCommand	Command:	66

User Permissions

admin

Group Permissions

Admins
Users

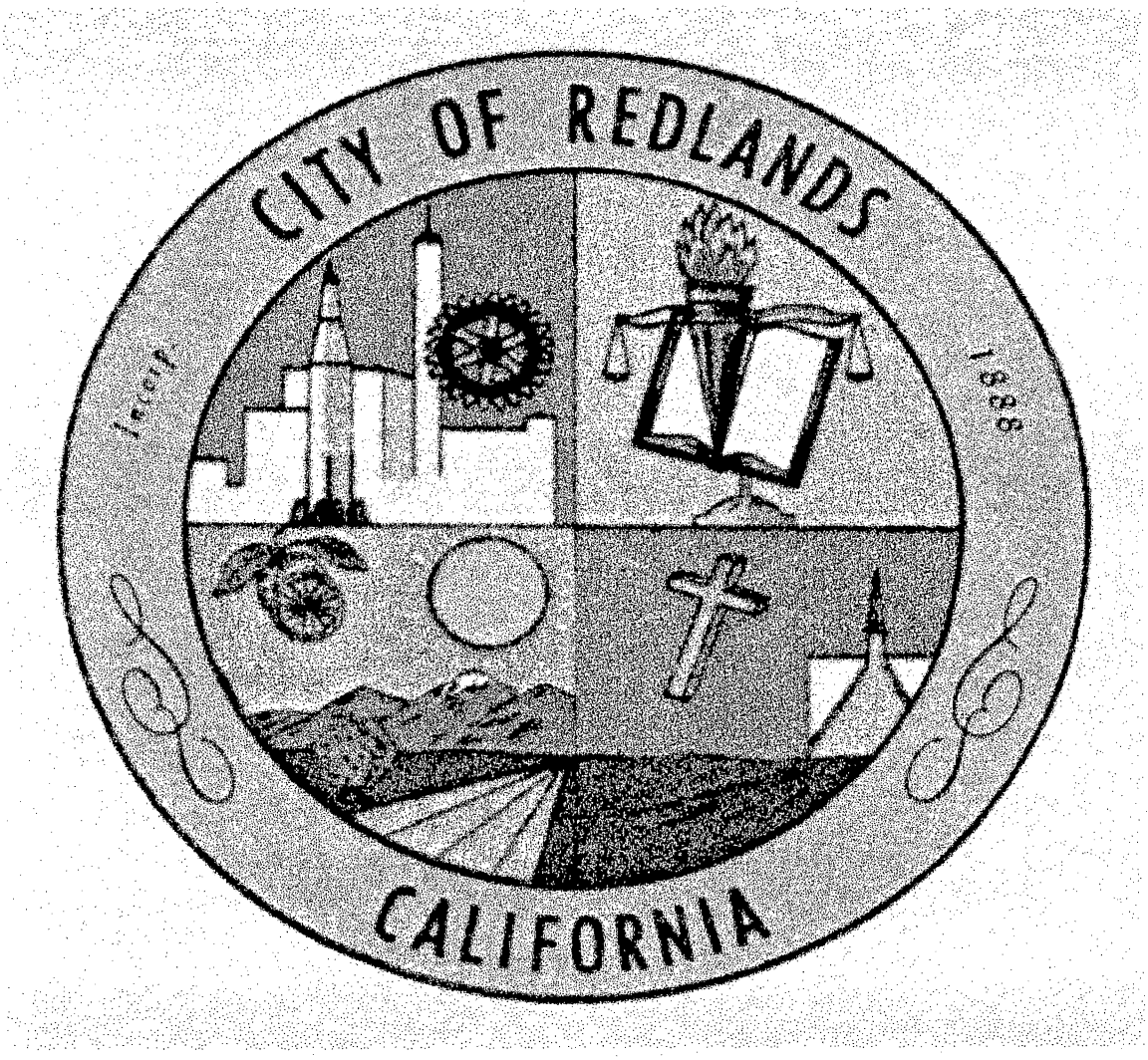
```
Option Compare Database          'Use database order for string comparisons

Sub Backup_Click ( )
On Error  GoTo Err_Backup_Click

        Dim DocName As String
        Dim LinkCriteria As String

        DocName = "BACKUP"
        DoCmd OpenForm DocName, , , , LinkCriteria

Exit_Backup_Click:
        Exit Sub

Err_Backup_Click:
        MsgBox Error$
        Resume Exit_Backup_Click

End Sub

Sub Daily_Call_Log_Click ( )
On Error GoTo Err_Daily_Call_Log_Click

        Dim DocName As String
        Dim LinkCriteria As String

        DocName = "Daily Call Log"
        DoCmd OpenForm DocName, , , , LinkCriteria

Exit_Daily_Call_Log_Click:
        Exit Sub

Err_Daily_Call_Log_Click:
        MsgBox Error$
        Resume Exit_Daily_Call_Log_Click

End Sub

Sub Division_Click ( )
On Error GoTo Err_Division_Click

        Dim DocName As String
        Dim LinkCriteria As String

        DocName = "Division"
        DoCmd OpenForm DocName, , , , LinkCriteria

Exit_Division_Click:
        Exit Sub

Err_Division_Click:
```

```
        MsgBox Error$
        Resume Exit_Division_Click

End Sub

Sub Exit_Click ( )
On Error GoTo Err_Exit_Click

        DoCmd Quit

Exit_Exit_Click:
        Exit Sub

Err_Divisions_Click:
        MsgBox Error$
        Resume Exit_Exit_Click

End Sub

Sub Print_Daily_Call_Log_Click ( )
On Error GoTo Err_Print_Daily_Call_Log_Click

        Dim DocName As String
        Dim LinkCriteria As String

        DocName = "Print Call Log"
        DoCmd OpenForm DocName, , , , LinkCriteria

Exit_Print_Daily_Call_Log_Click:
        Exit Sub

Err_Print_Daily_Call_Log_Click:
        MsgBox Error$
        Resume Exit_Print_Daily_Call_Log_Click

End Sub

Sub Print_Other_Reports_Click ( )
On Error GoTo Err_Print_Other_Reports_Click

        Dim DocName As String
        Dim LinkCriteria As String

        DocName = "Print Reports"
        DoCmd OpenForm DocName, , , , LinkCriteria

Exit_Print_Other_Reports_Click:
        Exit Sub

Err_Print_Other_Reports_Click:
        MsgBox Error$
        Resume Exit_Print_Other_Reports_Click
```

```
End Sub

Sub Update_Call_Log_Click ( )
On Error GoTo Err_ Update_Call_Log _Click

        Dim DocName As String
        Dim LinkCriteria As String

        DocName = "Update Call Log"
        DoCmd OpenForm DocName, , , , LinkCriteria

Exit_Update_Call_Log _Click:
        Exit Sub

Err_Update_Call_Log _Click:
        MsgBox Error$
        Resume Exit_ Update_Call_Log _Click

End Sub
```

A P P E N D I X F

USER MANUAL



PUBLIC WORKS
CALL LOG DATABASE
USER MANUAL

Designed by Linda Webster

TABLE OF CONTENTS

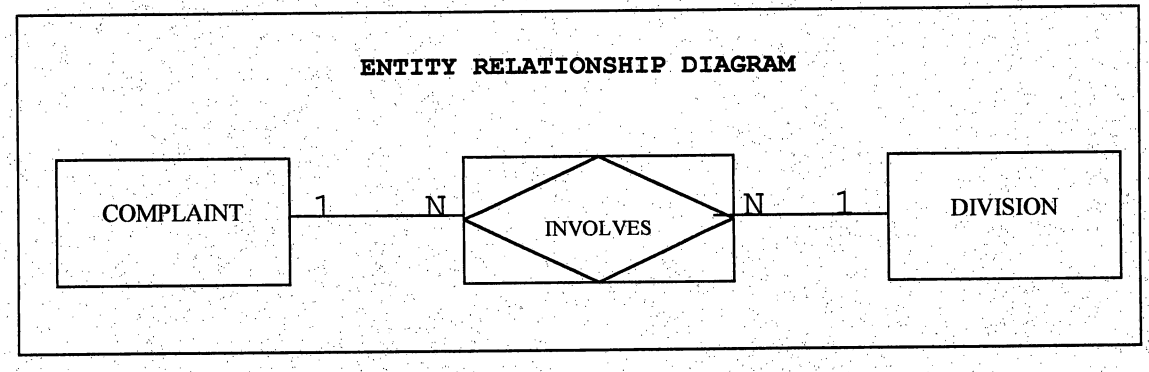
INTRODUCTION

The Call Log database was created in Microsoft Access® 2.0, a relational database management system for Microsoft® Windows™, permitting a graphical user interface, and a user-friendly work environment.

This manual will cover the Call Log database form views, and tasks that can be achieved through each. The Microsoft Access *User's Guide* and Microsoft Access *Building Applications* manual accompanying Microsoft Access 2.0 are referenced in this manual. All macros and event procedures used in this application are located in appendix B.

Relationships

The Call Log database contains three related tables. Each table represents and entity set. Each row in the tables represents an individual record. The following Entity Relationship Diagram depicts the table relationships.



There is a one-to-many relationship between COMPLAINT and INVOLVES, and a one-to-many relationship between DIVISION and INVOLVES.

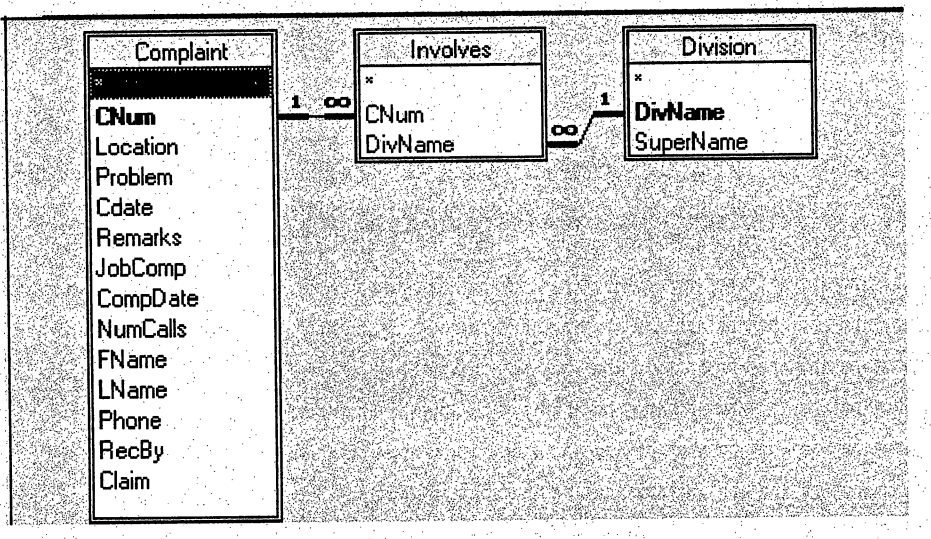
- COMPLAINT table: Contains information pertaining to specific calls.

- The INVOLVES table: Used to eliminate a many-to-many relationship between the COMPLAINT and DIVISION tables. It will maintain historic data on the divisions involved in specific complaints.

- Division table: Contains the division's complaints are track for, and the current superintendent of each division.

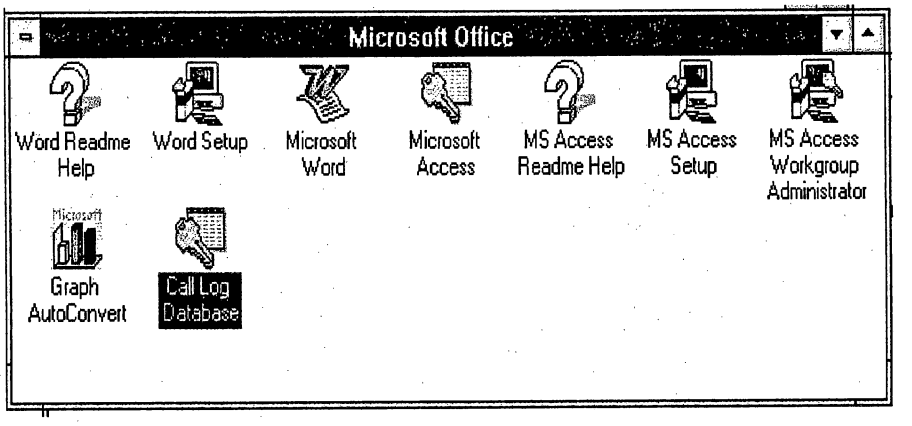
Table Structure

The following table structure is used in the Call Log database. For attribute (field) definitions see the Data Dictionary in Appendix A.



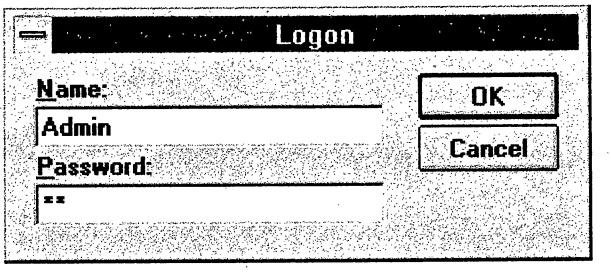
EXECUTING THE DATABASE

To execute the Call Log database double click the Call Log database icon in the Microsoft Office group, located in the Windows Program Manager window. Items in your Office group may differ from this example.



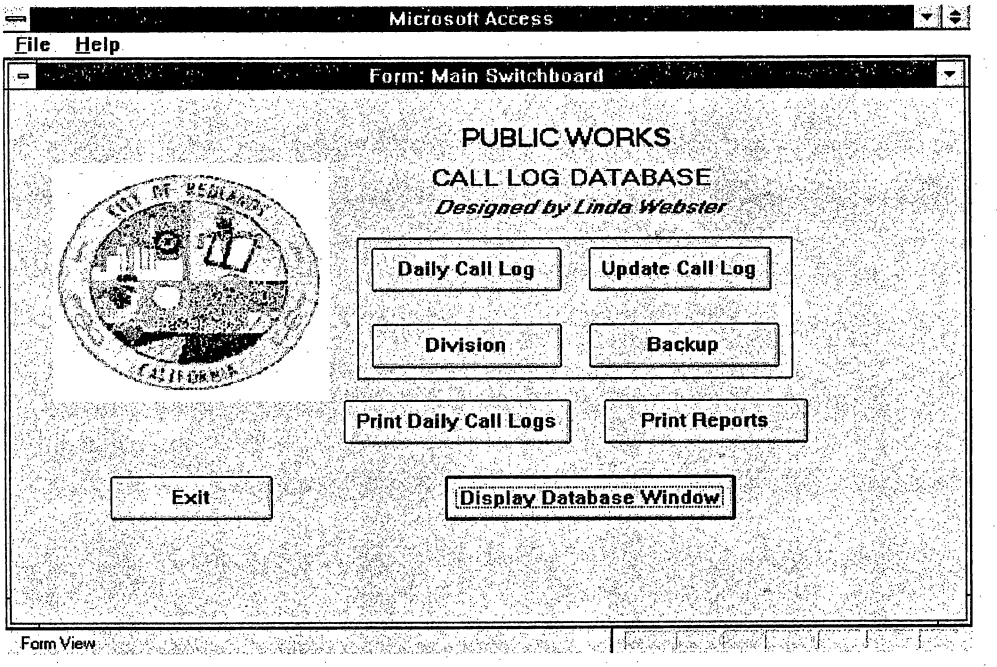
Logon

To access the database you must enter a logon name and password. When the logon window appears enter ***Admin*** in the Name text box and your password in the password text box. Click OK to continue.



THE MAIN SWITCHBOARD

Once you have logged on the Main Switchboard screen appears. This screen is used as the main access to the Call Log database.

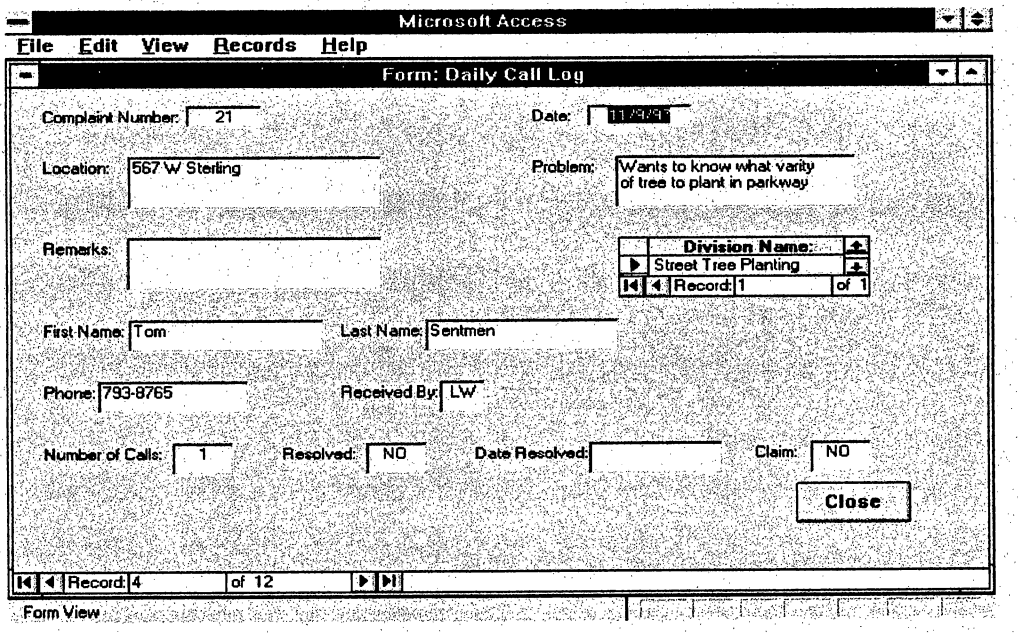


The Main Switchboard acts like a main menu in other applications. It allows you access to form views by clicking command buttons. The following forms and functions are accessed through the Main Switchboard.

- Call Log
- Update Call Log
- Division
- Backup
- Print Daily Call Logs
- Print Reports
- The Database Window
- Exit Microsoft Access

Adding New Records

New records will be added to the database using the Daily Call Log form. To access this form, click the Daily Call Log command button on the Main Switchboard.



The Daily Call Log form has a blank record, which follows the last existing record. If there are no existing records to display, the blank record is the only record. New data is entered in to the blank record.

You can reach the blank record by advancing one record at a time by clicking the right arrow button at the bottom of the screen, or clicking the arrow button with the "|", this button will advance you to the blank record.

If you want to add records and don't want to view existing records, you can use the Data Entry command. Only records you add are then available in the form view. Existing records are not available for view using the Data Entry command.

To Add Records Using The Data Entry Command

1. Choose Data Entry from the Records menu. The insertion point appears in the Location field of the form.

2. Type values for each field, press TAB to move to the next field.

3. After you fill in all the fields, press TAB, or use the right arrow button at the bottom of the form to move to the new blank record

When you move to the next record, Microsoft Access saves the record you added to the database. When you finish adding records, just close the form using the Close button. You do not have to save your work.

Required Fields

The following fields must contain data on the Daily Call Log form. That is, they can not be left blank.

- Complaint Number: This number is automatically generated.
- Date: This field is automatically generated and will contain the current date.
- Location: You must enter a location.
- Problem: You must enter what the problem is.
- Division: You must enter the division(s) name

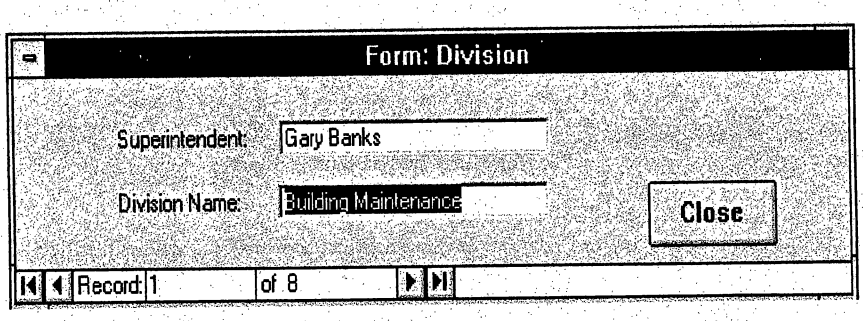
Fields Set To Default Values

The following fields contain default values on the Daily Call Log form. These values may be changed if necessary.

- Number of Calls: Default value "1".
- Resolved: Default value "NO"
- Claim: Default value "NO"

Adding a New Division

A new division can be added to the database through the use of the division form. To access this form, click the Division command button on the Main Switchboard.



1. Advance to a blank record. You can reach a blank record by advancing one record at a time clicking the right arrow button at the bottom of the screen, or clicking the arrow button with the "|", this button will advance you to the blank record.

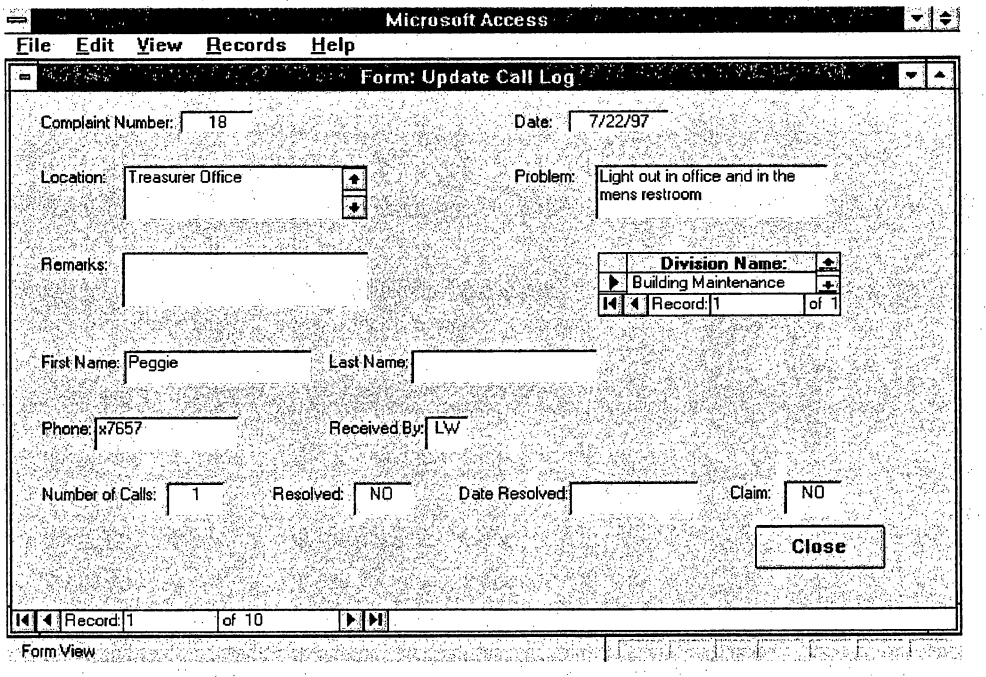
2. Enter the new division in the Division text box.

3. Enter the superintendent's name in the Superintendent text box.

When you move to the next record, Microsoft Access saves the record you added to the database. When you finish adding records, just close the form using the Close button. You do not have to save your work.

Updating Records

Updating records in the database will be done through the Update Call Log form. To access this form, click the Update Call Log command button on the Main Switchboard. To add new records use the Daily Call Log form.



When a complaint has been resolved and you want to update the record to indicate this, use the Find command to locate the complaint number.

The Find Command

1. Place the insertion point in the Complaint Number text box.

2. Select Find from the Edit menu.

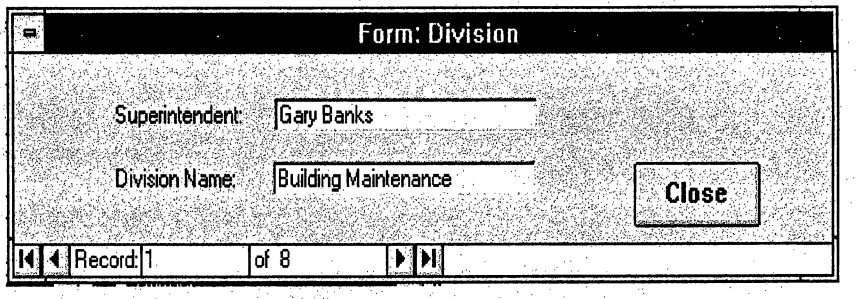
3. In the Find What text box enter the complaint number you want to find

4. Click the Find First button.

The Find command can be used with any field on the form. For more information regarding the Find command see *Microsoft Access User's Guide*, pages 84-90.

Updating Superintendent's Name

To change the name of an exiting superintendent use the Division form. To access this form, click the Division command button on the Main Switchboard.



To change the name of a superintendent who appears in one or more records use the Find and Replace command.

Find and Replace Command

1. Select Replace from the Edit menu.

2. Place the insertion point in the Superintendent text box.

3. Enter the name to be changed in the Find What text box.

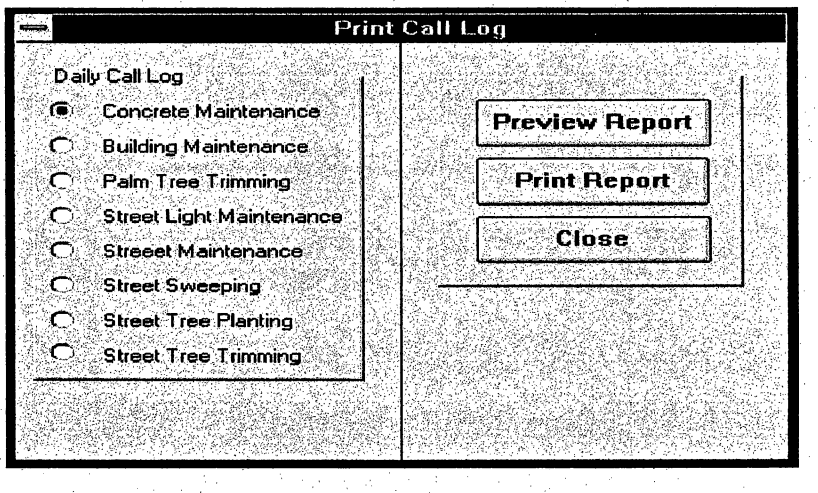
4. Enter the new name in the Replace With text box.

5. Click the Replace All button.

The name will be replaced in all locations. For more information regarding the Find and Replace command see *Microsoft Access User's Guide*, pages 87-90.

Printing Daily Call Log Reports

To print Daily Call Log report(s) click the Print Daily Call Log Reports command button on the Main Switchboard.



1. Select the report you want to print.

2. Click the Print Report button.

3. Enter the call date when prompted, click OK.

The report will then be sent to the printer

Preview A Report

To view a report on the screen.

1. Select the report you want to preview.

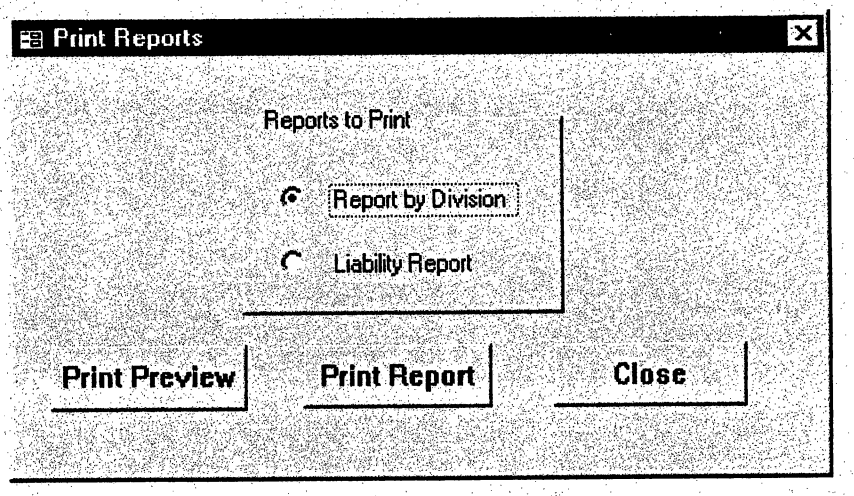
2. Click the Preview Report button.

3. Enter the call date when prompted, click OK.

The report will be displayed on the screen.

Printing Report by Division and Liability Report

To print the Report by Division and Liability Report click the Print Reports command button on the Main Switchboard.



1. Select the report you want to print from the Print Reports form.

2. Click the Print Report button.

3. Enter the beginning and ending date when prompted.

4. Click OK.

The report will then be sent to the printer

Preview A Report

To view a report on the screen.

1. Select the report you want to preview from the Print Reports form.

2. Click the Preview Report button.

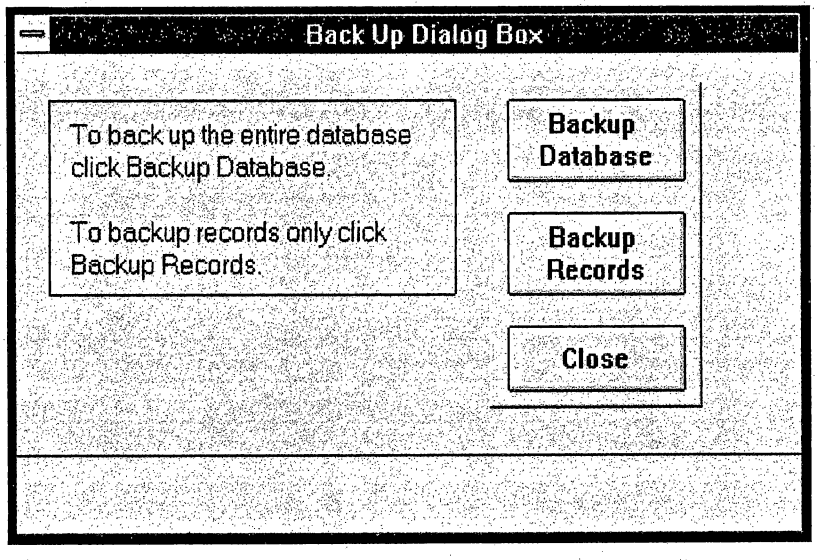
3. Enter the beginning and ending date when prompted.

4. Click OK.

The report will be displayed on the screen.

Backing Up The Database

The Back Up Dialog Box is used to backup the database. To access this form, click the Backup command button on the Main Switchboard.



To Backup The Entire Database

1. Click the Backup Database button.

2. You will be prompted to enter a disk in the A drive. Place the **BACKUP DATABASE** disk in the A drive.

3. Click OK.

4. Microsoft Access will warn you that the Make-table query will modify data. And ask if you want to continue.

5. Click OK to continue.

6. You will be prompted when the backup is complete.

Backing Up Records

To backup the completed records to the A drive and delete them from the hard drive:

1. click the Backup Records button.

2. You will be prompted to enter a disk in the A drive. Place the **BACKUP RECORDS** disk in the A drive.

3. Click OK.

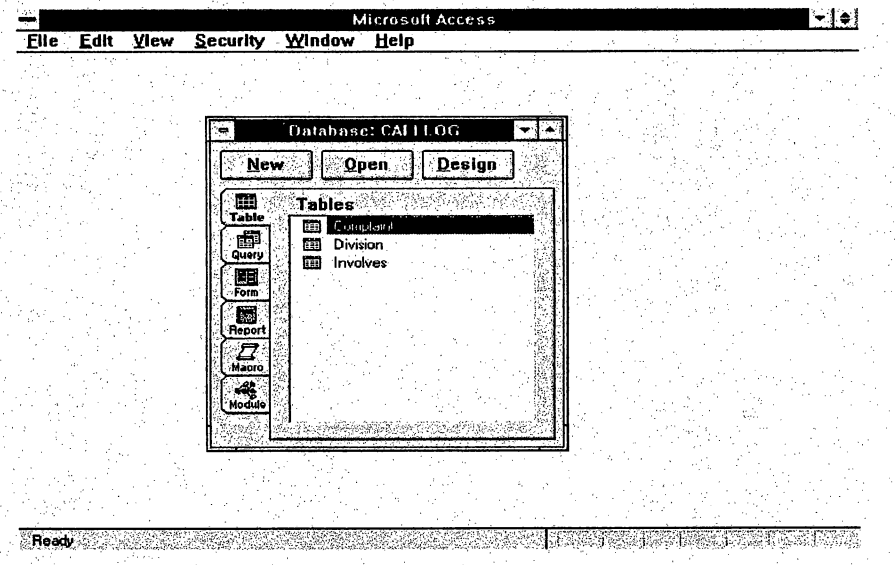
4. Access will warn you that the Append query will modify data. And ask if you want to continue.

5. Click OK to continue.

6. You will be prompted when backup is complete

THE DATABASE WINDOW

To access the database window click the Display Database Window command button on the Main Switchboard.



Creating Queries

You may want to view or display data in the database in another form other than the available predefined queries that accompany this application. You can do this by creating ad hoc queries.

To create a query you must be in the database window.

1. Click the query tab from the database window.

2. Click the New button.

3. Refer to Chapter 11 Designing Select Queries in the Microsoft *User's Guide* page 227, to create a new query.

Creating Reports

The predefined reports that are available in this application are the reports that you use most frequently. There may be a time when you need to create new reports.

To create a new report you must be in the database window.

1. Create a new query. See Creating Queries above.

2. Click the Reports tab from the database window.

3. Click the New button.

4. You must indicate the data source, click the ▼ arrow and select from the list.

5. Refer to Chapter 20 Report Basics, in the Microsoft *User's Guide*, page 485.

Passwords

When this application was installed you entered your password. For security purposes it is recommended that you change you password on a regular basis.

Changing Passwords

1. You must be in the database window.

2. Choose Change Password from the Security menu.

Refer to Microsoft *Building Applications* To change your password, page 338.

For other security measures refer Chapter 14 Securing Your Application in Microsoft *Building Applications* manual.

RESTORING THE DATABASE

In the case of a system failure data may be lost. If this happens you can restore the database from the **BACKUP DATABASE** disk. Remember that the data contained on this disk is from your last backup. Any new data entered from the time of the last backup to the time of failure will not be restored.

To Restore The Database

1. Reinstall Microsoft Access if necessary.

2. Insert the **BACKUP DATABASE** disk into the A drive.

3. Start Microsoft Access.

4. Select Open Database from the File menu.

5. Select the A drive from the Drive text box.

6. Select bkupdb.mdb form the file name window.

7. Click OK.

8. Select the Query tab from the database window.

9. Select and run the listed queries in the order they are listed.

10. Select Exit from the File menu.

11. Start the Call Log database.

If it is necessary to reinstall Microsoft Access you will need to set reactivate the Logon window. To do this, refer to Chapter 14, Securing Your Application in the Microsoft *Building Applications* manual.

ACCESSING OLD RECORDS

You can access the old records that have been removed from the database from the database window.

1. Select Open Database from the File menu.

2. Insert the **BACKUP RECORDS** disk in the A drive.

3. Select the A drive from the Drive text box.

4.Select oldrec.mdb form the File Name window.

5. Click OK.

To view the records

1. Click the table tab if necessary.

2. Select the Open button from the database window.

3. Record will be displayed in datasheet view.

Table: Old Complaints

Complaint_CNun	Location	Problem	Cdate	Remarks
15	Smiley Library	Tree limb down in t	7/18/97	
22	2945 Mill Creek Rd	limb down	12/31/97	
(Counter)				

You may print the records in datasheet view by selecting Print from the file menu. Or you can perform a query on the table for specific data. To create a query, see Microsoft *User's Guide* Chapter 11, Designing Select Queries page 227.

To return to the Call Log database

1. Select Open Database from the file menu.

2. Select the C drive from the Drive text box.

3. Select calllog.mdb form the File Name window.

4. Click OK

For more in-depth coverage of items in this manual or items not covered by this manual, refer to the Microsoft *User's Guide and Building Applications* manual.

A P P E N D I X A

DATA DICTIONARY

See Appendix B of paper for Data Dictionary.

A P P E N D I X B

MACROS AND EVENT PROCEDURES

See Appendix E of paper for macros and event procedures.

WORKS CITED

Rob, Peter and Carlos Coronel. Database Systems Third Edition, Cambridge, MA; Course Technology, 1997.

McFadden, Fred R. and Jeffery A. Hoffer. Database Management. Second Edition, Menlo Park, CA; Benjamin/Cummings, 1988.

Date, C. J.. Relational Database Writings 1985-1989, Reading, MA; Addison-Wesley, 1990.

"Mission Statement". City of Redlands

Mutter, Ron, Director. Public Works Department, City of Redlands; Phone Interview: January 8, 1997

Johnson, Lauri, Secretary. Public Works Department, City of Redlands; Personal Interview:
Date: January 16, 1997
Time: 4:00 p.m.
Date: March 10, 1997
Time: 1:00 p.m.
Date: July 16, 1997
Time: 8:00 a.m.
Date: November 12, 1997
Time: 8:00 a.m.
Date: January 6, 1998
Time: 8:00 a.m.