A MULTIPLE DATABASE MANAGEMENT SYSTEM: MDMS

By

JINGHUA LIU

Bachelor of Arts

Columbia College

Columbia, Missouri

1992

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
July, 1997

A MULTIPLE DATABASE MANAGEMENT

SYSTEM: MDMS

Thesis Approved:

H. Lu

Thesis Adviser

J. Chandler

Jacques LaFrance

Thomas C. Collins

Dean of the Graduate College

ACKNOWLEDGMENTS

I wish to express my appreciation to my major advisor, Dr. Huizhu Lu for her supervision and guidance throughout this study. My appreciation extends to my other committee members Dr. John P. Chandler and Dr. Jacques LaFrance for all the help and support they have given to me. I would like to thank Dr. John P. Chandler and the Computer Science Department for providing me the opportunity of teaching assistantship.

I would also like to give my special appreciation to my wife and parents, their support, understanding and encouragement always along with me throughout the years of my study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

A global decision or planning usually relies on information stored in independently developed databases. Because both operation and data types supported by local databases are vary from one DBMS to another. The conventional approach of database management system can not support increasing global information need. This problem raises the issue of multiple database management. In the decade, many approaches and implementations have been provided to solve this problem. This thesis presents MDMS(Multiple Database Management System) as an attempt to simulate a multiple database management system.

Motivation

By the design and implementation of MDMS, the author attempts to simulate a multiple database management system that combines the multiple database accessing functionality with the system management functionality including data source, application and user management. An object oriented approach is applied for MDMS design and implementation.

Objectives

The objective of the thesis is to introduce and simulate a multiple database management system. This system interacts with the users and the system administrator through graphical user interface. It supports users to run applications that access information stored in heterogeneous databases. It also supports the system administrator to manage system user, application and data source.

Organization of the Thesis

This thesis is composed of 6 chapters. The current chapter introduce the thesis topic MDMS, the motivation to develop it and its objectives.

Chapter II presents a review introducing the conventional database management system, global information need, the problems of the conventional database management system and the approaches proposed to solve this problems.

Chapter III introduces the object oriented system design of MDMS. It discusses the system objects, their features and functionality.

Chapter IV illustrates the system implementation of MDMS. The issues discussed in this chapter include class, MFC and inheritance.

Chapter V presents a complete description of MDMS graphical user interface design and the system functionality supported by GUI.

Chapter VI introduces the application installed in MDMS. The contents include data sources, data calculation and result tables.

Chapter VII is the conclusion. In this chapter, the advantages and possible future system extension is discussed.

The appendix presents the source data stored in Microsoft Access database and Oracle database.

CHAPTER II

LITERATURE REVIEW

In this section, the conventional database management system and the difficulty for it to meet global information need are discussed. An example is given to illustrate how this difficulty hinders people from making a global decision. The approaches to solve this problem and a skeleton of the MDMS are introduced.

Conventional Database Management System and Global Information

In conventional database management systems, applications rely on information stored in independent databases. These databases are typically developed to serve particular uses, such as a personnel or an inventory database that satisfies the managerial need of an enterprise. Over the past decades, both information technology and the information needs have been changed tremendously. Conventional information management systems can no longer meet increasingly sophisticated information requirements. An application, like a global decision support or planning, usually requires information from multiple, incompatible local data sources. The following example will introduce how a global decision support application can not rely on conventional database management system.

Example of A Decision Support

After a hurricane, many people claim insurance for their property damaging. The insurance company wants to estimate the amount of money they have to pay to these clients. This will decide whether the company needs to borrow money from bank and how much money to borrow if it is needed. To do so, the company needs to know which states, counties and cities are mopped up by the hurricane. If the company relies on conventional information management system, it would be difficult to find any information about the area covered by the hurricane from the insurance database. This information must be retrieved from a weather information database and sent to the insurance company, many efforts and time may have to be devoted to convert data defined by foreign DBMS to a proper type that is recognizable to the insurance database. Such process is obviously not desirable or unacceptable to today's world in which hundreds of decisions can be made in a second.

Multiple Database Related Approaches

Previous example illustrates the difficulties encountered when people try to make global decision that needs information stored in those independent, heterogeneous, local data sources. In the decade, many efforts have been made to solve this problem. The concepts and architectures of heterogeneous databases [Reddy, 1992] and multi-databases [Bright, 1994] were proposed to provide integrated global access to heterogeneous local databases. The system architectures proposed are either a front end to multiple local DBMSs or a global system layer on the top of local DBMSs. In heterogeneous databases, semantically similar data may have different names and data types defined by local DBMSs. Many theories and models of semantic integrity are proposed to solve the conflict that occurs due to this inconsistency [Sciore, 1994]. From a more generic view of data source, Gio Wiederhold proposed the concept and architecture of mediator. As an intermediary layer between end users and diverse data sources, mediator provides data accessing, abstracting and integrating services [Wiederhold, 1995].

The next section introduces MDMS, a Multiple Database Management System. This system supports multiple database application and system management functionality.

The MDMS

The MDMS is composed of five modules: main program, application, database,

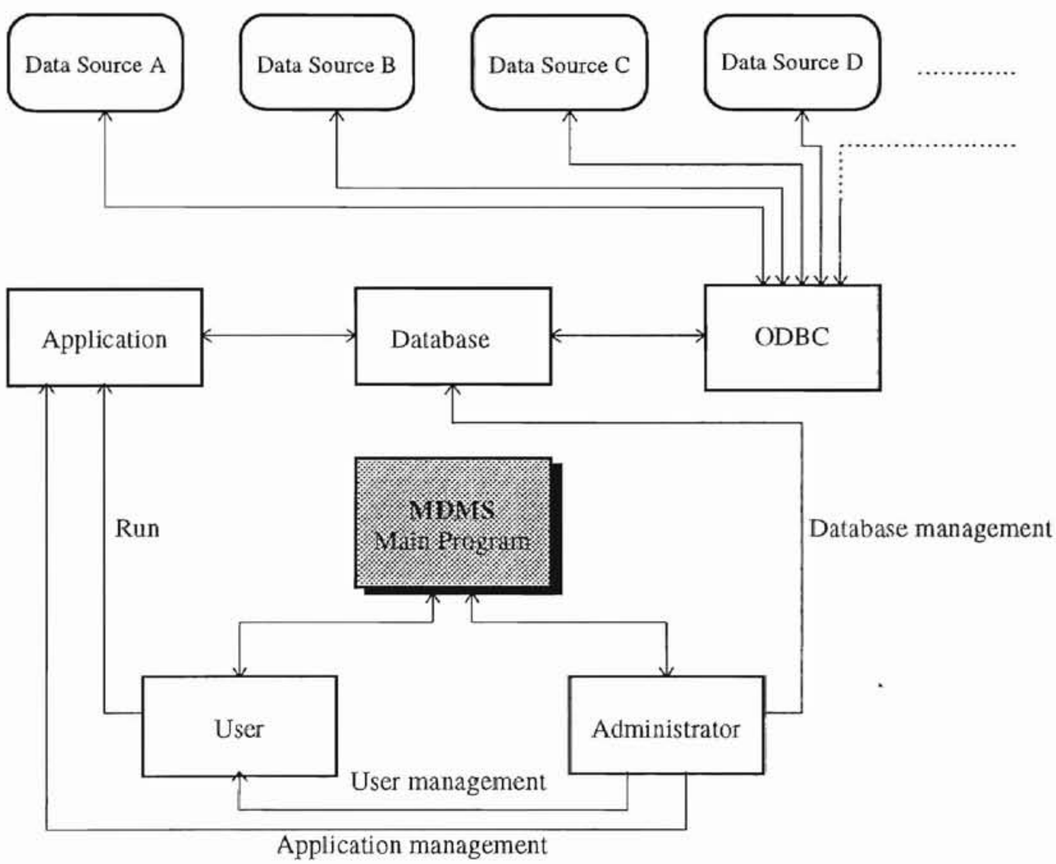


Figure 1. A Skeleton of MDMS

user and administrator. Figure 1 illustrates the skeleton of MDMS. The main program provides an entrance of the system. It interacts with user and administrator through the user interface and administrator interface windows respectively. The user can run application that relies on information stored in multiple, heterogeneous data bases. The system administrator can perform system management tasks. The details about ODBC and these modules will be discussed in Chapter III and Chapter IV.

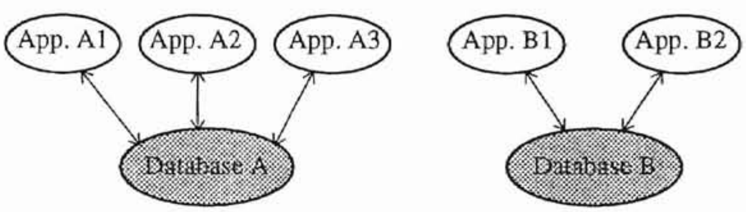
CHAPTER III

SYSTEM DESIGN

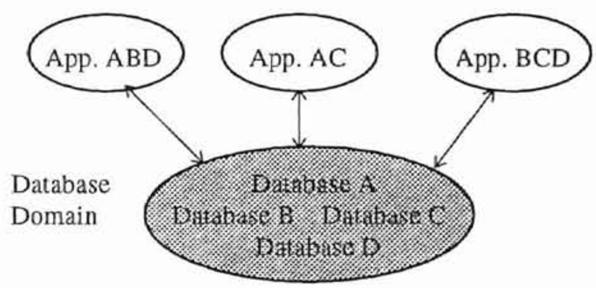
To design and implement the system, the first step is to analyze the overall system goal. The goal of target system is to support applications that access information stored in diverse databases. This system is managed by system administrator and allows end users to get information they need by running applications. In this description of system goal, some objects emerge and each plays a role in overall system function. These objects are application, database, user and administrator. Each of these objects has its own identity and functionality, they communicate each other to achieve the overall system goal. Therefore, we start system design and implementation from discussion of objects application, database, administrator and user.

Database Object

In a multiple database management system, applications usually access information stored in multiple, heterogeneous databases. Each application tells the system what information it needs and which database contains this information. The system must possess the way to access all the particular databases according to the request. On the other hand, the system supports multiple applications and each application may requires information from different sets of data sources. The system has to provide a common, single interface between applications and heterogeneous data sources. To achieve this goal, the database object must be a domain of heterogeneous databases instead of particular databases. Figure 2 describes the difference between a multiple database object and a conventional one respect to the relationship between database and application. The relationship between conventional database and application is illustrated in part (a), App. A1, App. A2 and App. A3 are applications that rely on information stored in database A. While App. B1 and App. B2 are applications using data stored in database B. Because these application are developed based on the data defined and managed by particular DBMS, none of the applications rely on database A can access data stored in database B



(a) Conventional Database and Application



(b) Multiple Database and Application

Figure 2. Databases and Applications

or vice versa. In part (b), each of three applications requires information through a database object that supports access to database A, database , database C and database D. Application App. ABD relies on information stored in databases A, B, D; Application App. AC uses data stored in database A, C; Application App. BCD requires information stored in database B, C and D. To provide the means of accessing to heterogeneous databases, the database object must be able to communicate with diverse DBMSs no matter what platform any particular DBMS relies on or what communication protocol it uses to send and receive information. It also needs to solve the data type definition conflict produced by independently developed local DBMSs.

Application Object

A database application is a program that conveys user's query to target database and receives expected result if the query is successful. The difference between an multiple database application and a conventional one is the former plays queries on heterogeneous databases instead of a particular one. The target system supports multiple applications and each of them requires information from a different subset of the domain of database object. Therefore, the application object contains a domain of applications and it has a many to many relationship with database object. This relationship is described in Figure 3. In object oriented programming, objects communicate each other through message passing. Before an application program starts to handle data, two messages need to be passed from application object to database object. One indicates what database the current application wants to access. Another indicates what operation will be performed on that database. The problem arise here is the inconsistency existing among the independently developed databases. The way to connect database, data type definition and SQL language vary from DBMS to DBMS. To build a common, single interface between applications and databases, a standard to bridge the gaps among heterogeneous databases must be established.

User Object

A user of multiple database management system has no significant difference from a user of conventional database management system. This is due to the fact that both systems hide database accessing details from users. The user doesn't need to know what database is going to be connected and how to get data from that database. He only deals with applications, that is, to run application program and get the result. Because most information stored in databases are confidential, a user must have login name and password to identify himself when he login the system. For same reason, not all of the

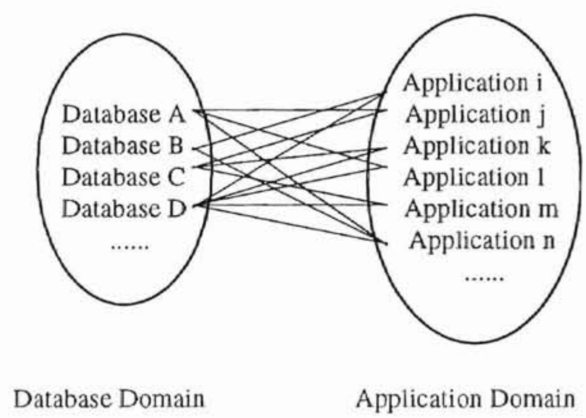


Figure 3. The Relationship between Database Object and Application Object

applications in the system are accessible to all system users, this restriction is established through the ownership. That means, a user only can access those applications owned by him.

Administrator Object

System administrator performs system management tasks including user management, application management and data source management. He or she can add or delete user, application or data source to keep the system updated. The administrator has his or her own login name and password to login the system. Unlike a user who has only a simple relationship with application, administrator has relationship with all the other objects.

CHAPTER IV

SYSTEM IMPLEMENTATION

In object oriented programming, an object is an instance of a class. A class catches the common property and behavior to identify its instances. According to the discussion in section 2, the target system has objects database, application, user and administrator. To implement these objects, we can either define new classes or inherit from existing ones. Microsoft Foundation Class (MFC) is a programming interface to windows, it contains a rich collection of basic window classes and becomes the most important component of Visual C++. Because Visual C++ 4.0 is the programming tool used to implement this system, all the classes defined in this system are derived from MFC classes.

Database Class

The database object has been discussed in previous section. To build a common, single interface between applications and databases, a standard to bridge the gaps among heterogeneous databases must be established. Microsoft ODBC and CDatabase, CRecordSet classes provide a solution to solve this problem. ODBC (Open Database Connectivity) is a database application development tool that supports application to access heterogeneous databases. To solve the data type conflict problem, two sets of data types are defined in ODBC, one is ODBC SQL data types and another is ODBC C types. When ODBC is used, any data types defined by local DBMS are mapped to equivalent

ODBC SQL types such that ODBC can use its SQL statement to access these data. Before these data are fetched into result set, they are mapped to equivalent ODBC C data types such that these data can stored by application program. Table 1 illustrates the conversion between ODBC SQL data types and ODBC C data types. ODBC is composed of ODBC drivers, driver manager and data sources. The most important component is ODBC drivers, they are mainly developed by database providers. The drivers implement ODBC function calls, manage communication protocols between application and data source, plays data query on target DBMS and return the result to application. To support database

ODBC SQL Data Type	ODBC C Data Type
SQL_CHAR	SQL_C_CHAR
SQL_VARCHAR	SQL_C_CHAR
SQL_LONGVARCHAR	SQL_C_CHAR
SQL_DECIMAL	SQL_C_CHAR
SQL_NUMERIC	SQL_C_CHAR
SQL_BIT	SQL_C_BIT
SQL_TINYINT(signed)	SQL_C_STINYINT
SQL_TINYINT(unsigned)	SQL_C_UTINYINT
SQL_SMALLINT(signed)	SQL_C_SSHORT
SQL_SMALLINT(unsigned)	SQL_C_USHORT
SQL_INTEGER(signed)	SQL_C_SLONG
SQL_INTEGER(unsigned)	SQL_C_ULONG
SQL_BIGINT	SQL_C_CHAR
SQL_REAL	SQL_C_FLOAT
SQL_FLOAT	SQL_C_DOUBLE
SQL_DOUBLE	SQL_C_DOUBLE
SQL_BINARY	SQL_C_BINARY
SQL_VARBINARY	SQL_C_BINARY
SQL_LONGVARBINARY	SQL_C_BINARY
SQL_DATE	SQL_C_DATE
SQL_TIME	SQL_C_TIME
SQL_TIMESTAMP	SQL_C_TIMESTAMP

Table 1. Conversion between ODBC SQL Data Types and ODBC C Data Types

application programming, MFC provides CDatabase class and CRecordSet class. CDatabase class creates a programming interface between application program and

ODBC, an object of this class connects an application program to a target database and then class member functions ExecuteSQL() is called to execute SQL statement against the database. An object of CRecordSet class represents a set of records selected from target data source. The Open() member function of this class selects data from data source, then the data member and the other member function can be called to perform desired operations on these data. Figure 4 illustrates how ODBC, CDatabase and CRecordSet classes are used when an application program accesses a particular database.

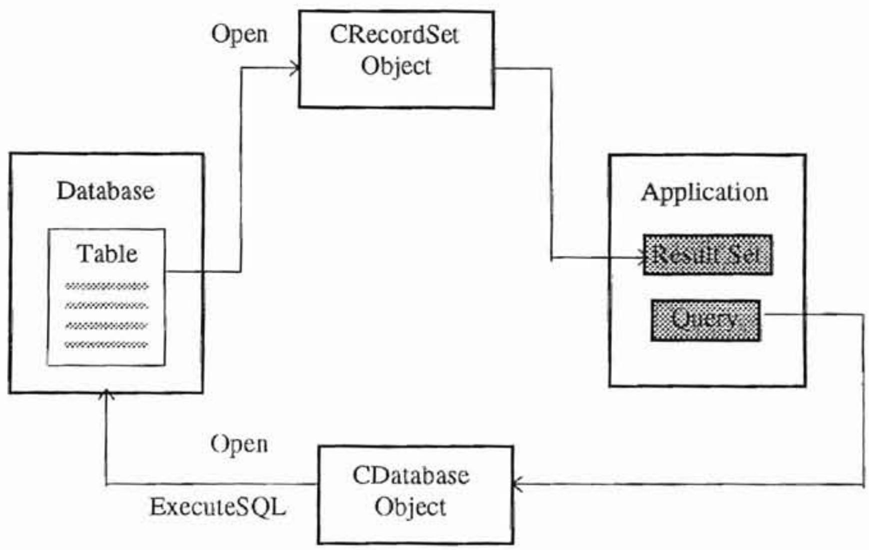


Figure 4. CDatabase and CRecordSet Classes in A Database Access

The previous discuss results the class used to implement the database object of the target system. This class is CMdmsDatabase derived from CDatabase base class, the prefix "Mdms" indicates that this class belongs to Multiple Database Management System - the target system. Besides this class, a record set class CMdmsRecordSet is derived from CRecordSet base class. The objects of class will be used to handle record sets selected from data sources. Because both CMdmsDatabase class and CMdmsRecordSet class are derived classes, they inherit all the properties and functionality of the base class CDatabase and CRecordSet.

Application Class

In Microsoft Foundation Class library, the class of application object is CWinApp. This class provides communication between applications and windows. For the target system, the main functionality of the application object are storing and running application programs. The class of this object can be derived directly from CWinApp. The derived class is CMdmsApp, object of this class serves as a library of application programs. When it interacts with a user, a class member function is triggered to run a selected application program. When it interacts with the system administrator, proper class member function is evoked to add or delete an application program.

User Class

In the target system, the functionality of user object is relatively simple. After a user login the system, all he does is choose and run application programs then get the results. Therefore, a proper class of user object should not only provide the support for running and storing application program but also provide a good dialog interface between user and application library. The most popular dialog class in MFC is CDialog. This class supports dialog box windows that contain a great variety of controls including button, edit box, list box, combo box, process bar and etc. When a user acts on any of these controls, the corresponding window message is sent to program and trigers the desired response.

These features make CDialog class the best choice, and the derived class for MDMS user object is CUserDlg.

Administrator Class

The system administrator performs multiple system management tasks mainly including user management, application management and data source management. This requires not only a good dialog interface but also a efficient way to switch from task to task. In MFC library, the class supports this feature is CPropertySheet. A property sheet is composed of task pages, each page can provide dialog box feature and indexed by a tab. When it opens, a property sheet displays the first page and all page tabs with task name on it. If a user want to switch from the current task page to another, all he needs to do is clicking on the tab of the target page. Therefore, MDMS uses property sheet as the administrator interface window and derive the corresponding class CAdminSheet from CPropertySheet.

CHAPTER V

GRAPHICAL USER INTERFACE DESIGN

MDMS graphical user interface is composed of program driver, login window, user interface window, administrator interface window. Figure 5 shows a layout of MDMS graphical user interface. In this section, each of these components will be discussed in details. In Visual C++ programming environment, an object not only has its class to define and implement its properties and functionality but usually also has its window representation to interact with users. The functionality of the object are implemented by the class member functions and these functions in turn are tightly related to window events. The communication between program and windows is established through the means of message mapping. When a window event takes place, for example, a button is clicked, the corresponding window message is immediately sent to the program and a corresponding member function is triggered to take response to this window message. This relationship between window and program binds the system user interfaces and functionality together such that it is difficult to illustrate each of them separately. Therefore, both graphical user interface design and system functionality are discussed in this section.

Program Driver

The functionality of program driver are opening MDMS starting window,

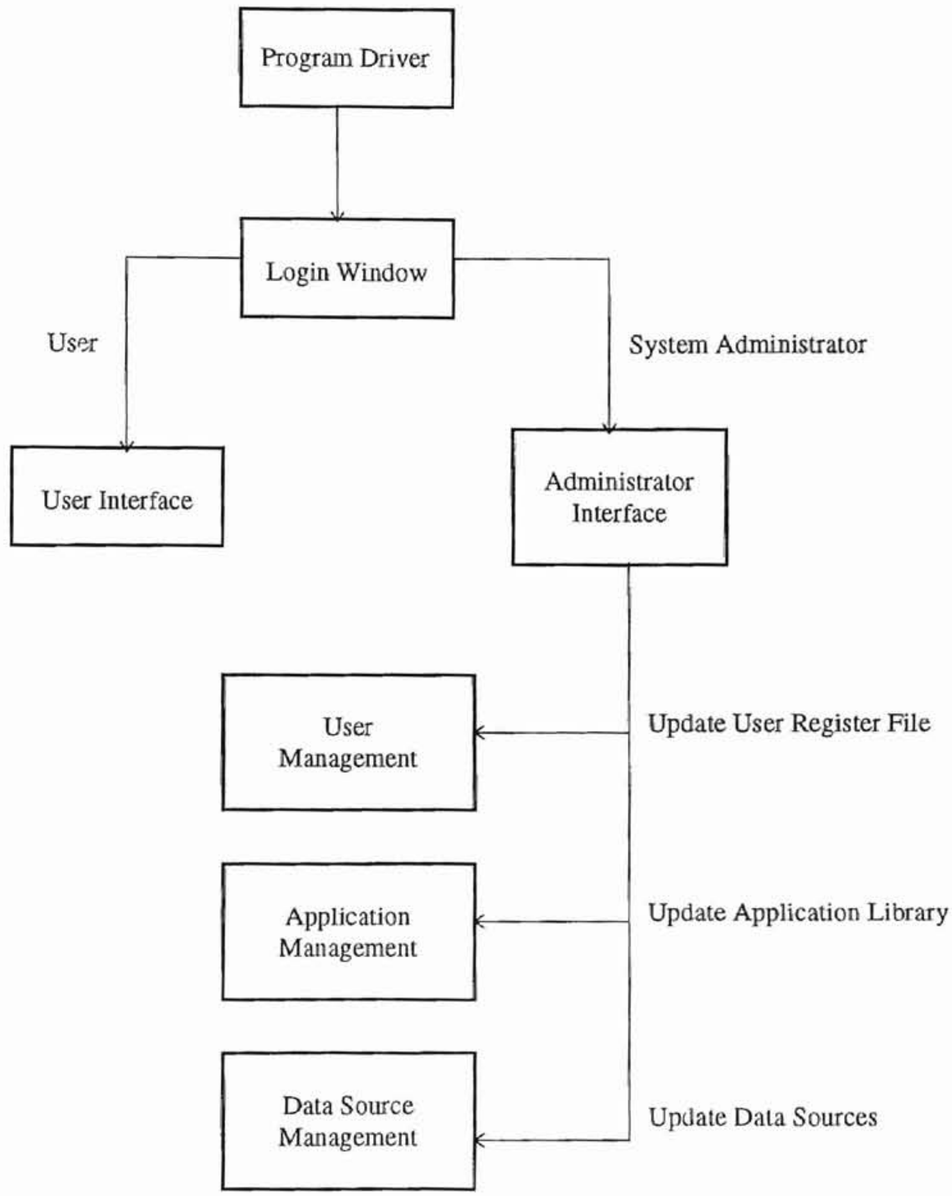


Figure 5. Layout of MDMS Graphical User Interface

recompiling /linking programs and running MDMS executable. Figure 6 shows the dialog window of program driver. There are three buttons on this window. Clicking on "About" button will open a help file window. The contents of this window introduce the system components of MDMS, its system files and system environment. Figure 7 shows the help file "About MDMS". If a user want to quit the system before open it, he can click on

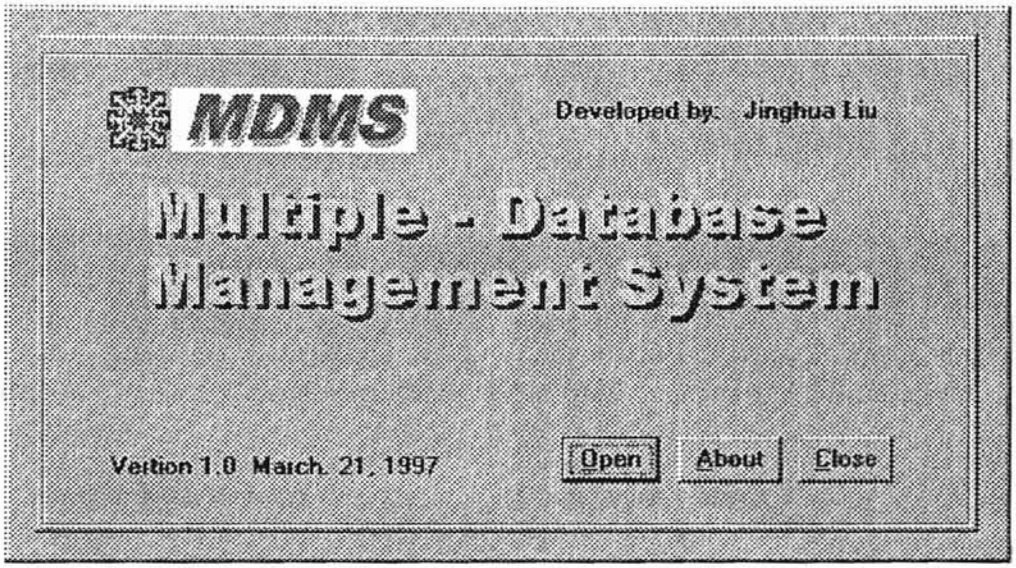


Figure 6. MDMS Program Driver Dialog Window

"Close" button. This will terminates MDMS starting window. When the "Open" button is clicked, two ShellExecute () function calls will be triggered and each of them executes a NMAKE command. NMAKE is a DOS command that runs Microsoft Program Maintenance Utility (NMAKE.EXE). When this command is executed, related source files are recompiled and linked to build a target project. For MDMS, the two NMAKE build application library and main program. The reason to rebuild these projects is that three program files will be modified whenever an application is added to or delete from the system. Two of them are app_lib.h and app_lib.cpp, they are source files used to build application library project. Another is UserDlg.cpp that is one of the source files of main program project. If these projects remain unrebuilt after a modification, the system will ask a user to rebuild them when he try to start MDMS again. After the projects are rebuilt, ShellExecute () function is called again to execute MDMS executable mdms.exe and a login window will show up.

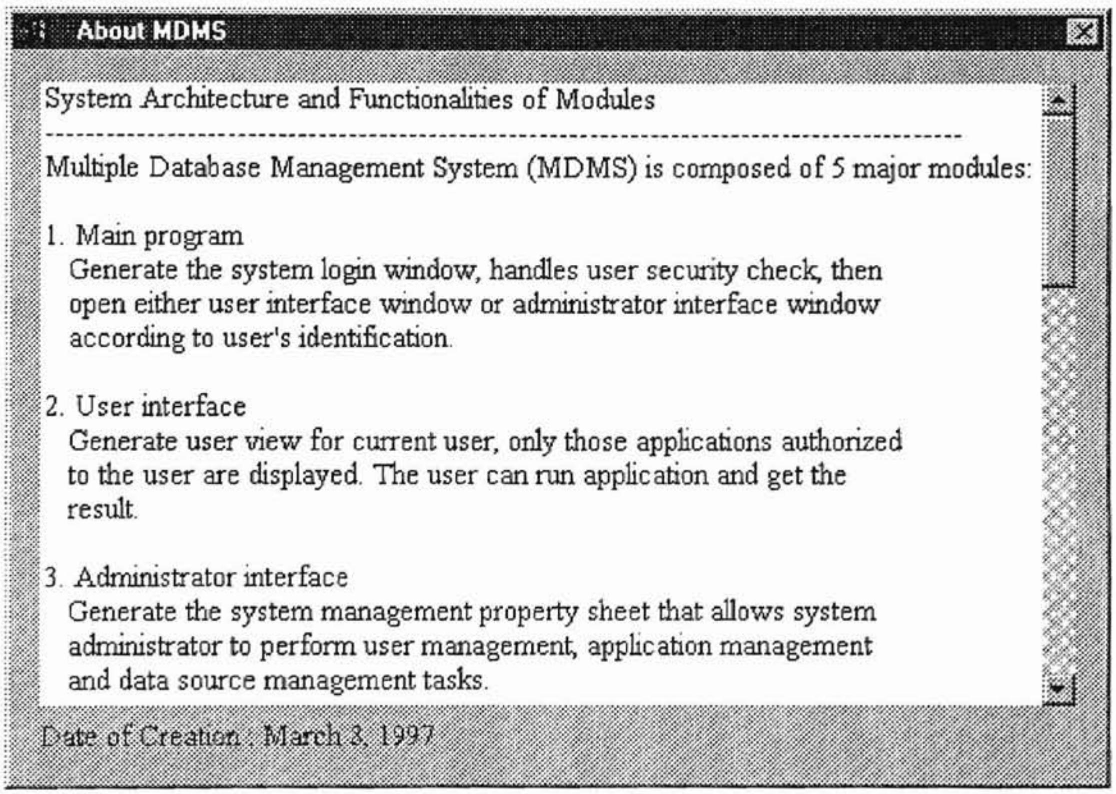


Figure 7. About MDMS Help File

Login Window

Figure 8 illustrates MDMS login window. This window has two edit boxes, the top one allows user to enter his login name and the lower one allows user to enter his password. After entering login name and password, one can either click on "OK" button or just simply hit return key and any of these will open a proper window according to the identification of the person. If the person is a user, MDMS user interface window will open for him. If the person is the system administrator, the window comes up is MDMS administrator interface window. The system identifies user and administrator by both their login name and password. The system administrator's login name and password are

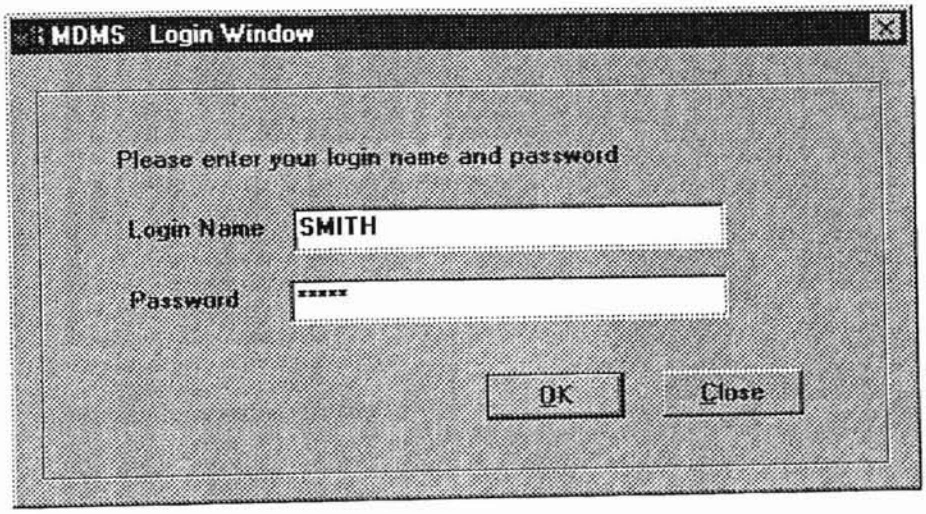


Figure 8. Login Window

hard coded in program. This name and password, certainly, is the only one can be used to open the administrator interface window and it can be changed only by manually changing the code. The system maintains a user register file "users" in directory c:\mdms\sys. This file contains all the login name and password of authorized users. It only can be updated by the system administrator and this will be discussed in section 5.4. If the login name or password entered can not match any login name and password stored in user file, the system displays a message box indicating that the login name and password are invalid. The person has to enter his login name and password again if he is an authorized user. Otherwise, he has to quit the system.

User Interface Window

MDMS user interface window provides the means of interaction between a user and MDMS application library. This user interface window is illustrated in Figure 9. The

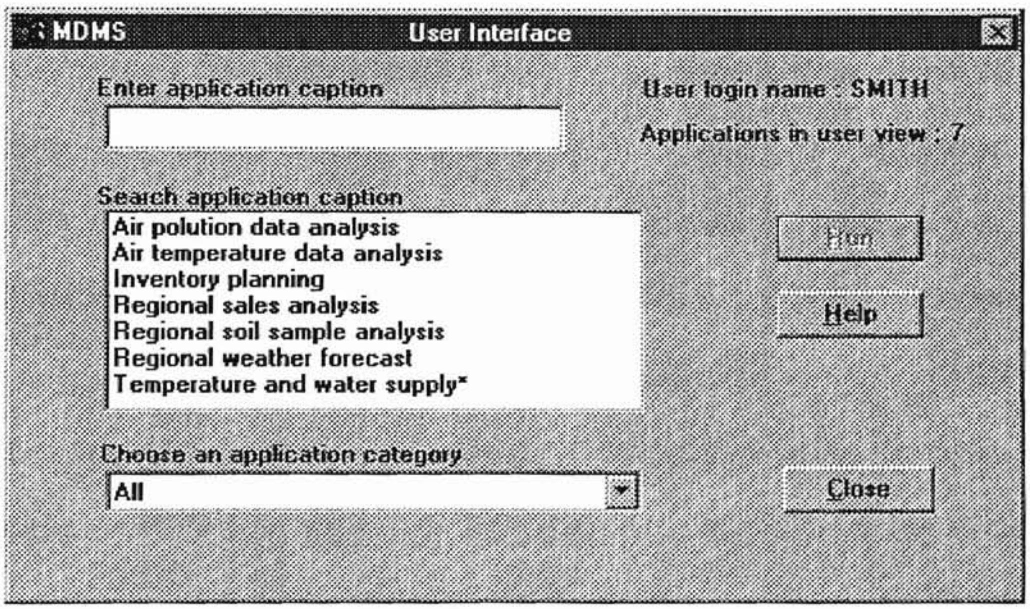


Figure 9. User Interface Window

information displayed on upper right corner of the window is user view information. It shows both the login name of the current user and the number of applications in current user view. A user view is a sub set of all applications stored in MDMS, it consists of only those applications owned by current user. The current user has no access to the other applications, this is another level of system security restriction enforced by MDMS. There are 7 applications displayed on the list box of the window. These applications are designed only for demonstration purpose and there could be much more applications in real world. To create an efficient and flexible tool for user to access applications, this window provides various ways of running application. The most direct way is entering the caption of an application in the edit box then clicking on "Run" button to run it. If the caption entered doesn't match any caption available, the system will display a message box indicating that the application entered doesn't exist. If the user can not remember what the caption exactly is, he can search through the list box to find the caption he wants then double click on that caption or click "Run" button to run it. If the caption displayed are too many or the current user still not sure which caption should be chosen, the application catalog combo box can be used to narrow down the searching area. When drop down arrow of the combo box is clicked, a drop list attached to the combo box is put down and the application categories show up. All applications currently in MDMS are classified into three categories: environment, weather and production. In Figure 9, the application catalog combo box shows "All". It means to display all the applications in current user view, this is the system default mode. After the user choose one of those categories from the drop down list, the system will update the list box display to reflect the user's choice. Figure 10 shows the result after user chooses "Environment" category. This time only three applications displayed on the list box. This will help the user to find an application if he remember what category the application falls into. And it also saves time may be used to search through a large caption list. All the details about how to

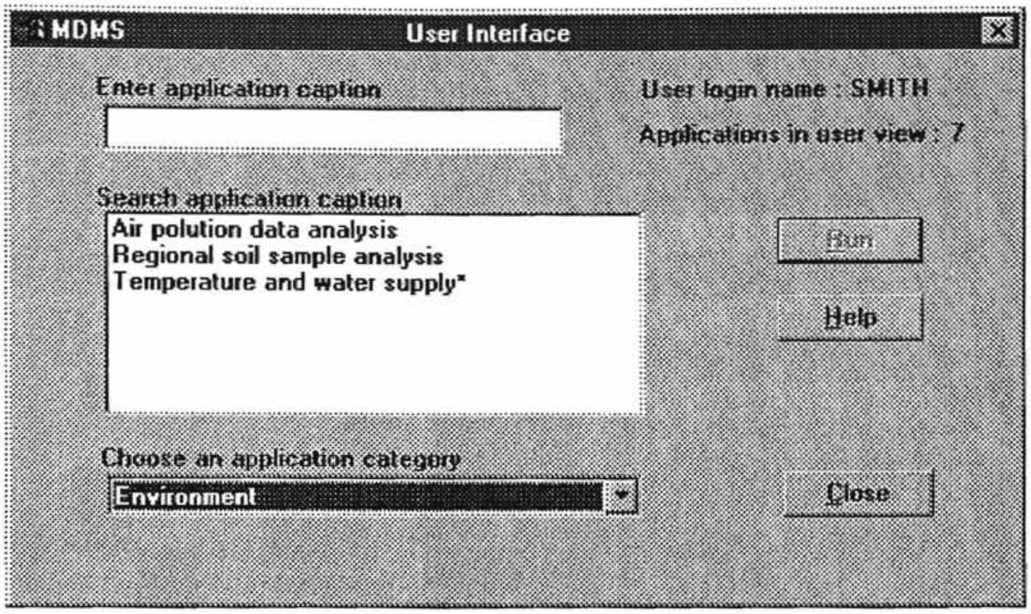


Figure 10. After Selecting An Environment Category

search and run applications are written in a help file. A user can click "Help" button to read this file.

Administrator Interface Window

Through administrator interface window, the system administrator interacts with system files and programs. This window is a property sheet composed of five task pages, each supports a particular system management task. These tasks pages will be introduced in following sections.

Add Application

Figure 11 shows the top page of the property sheet. Using this page, system administrator can add a new application to the system. There are two edit boxes on top of

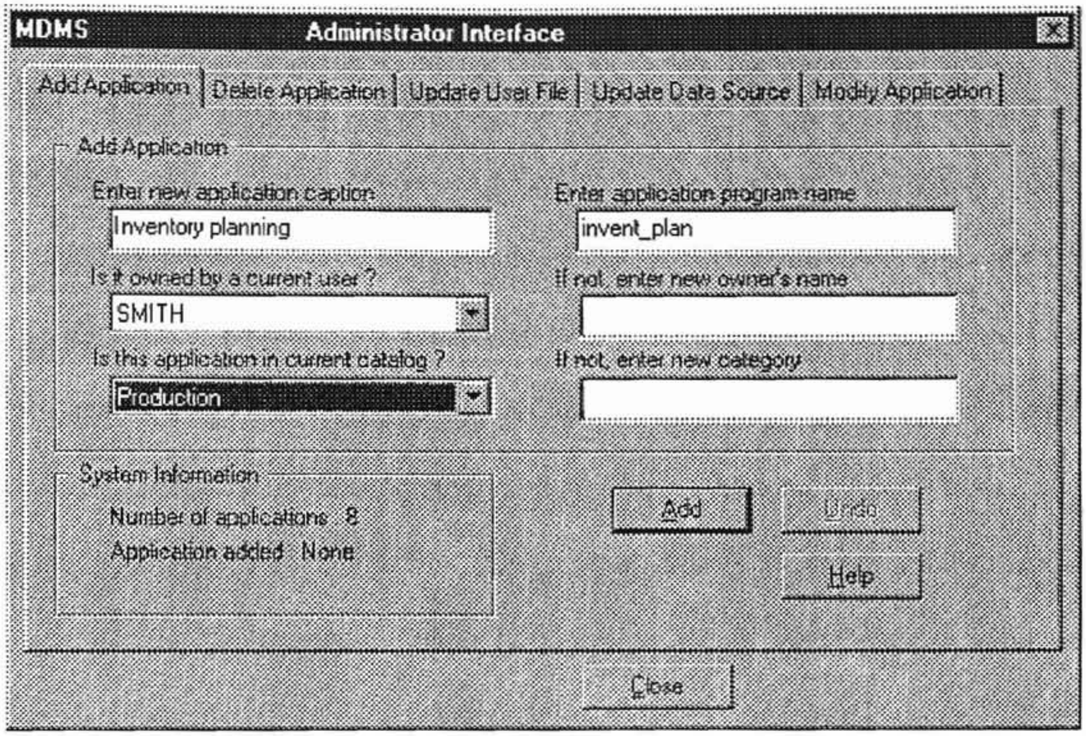


Figure 11. Application Adding Page

this page. To add a new application, the administrator needs to enter its caption in the first edit box. This caption normally is a short sentence simply telling what this application does. Next, the administrator enters the application's program name in the second edit box. This program does the real work of the application. If the new application is owned by a current system user, the administrator can find the user's login name on owner combo box. Otherwise, he has to enter a new owner name in the edit box next to owner combo box. If the new application falls into a current application category, the administrator can find the category from catalog combo box. Otherwise, he has to enter a new category in the edit box next to catalog combo box. In Figure 11, the given caption is "Inventory Planning", the program name is "invent_plan", the owner is "SMITH" and the category is "Production". The system information indicates that the current number of applications is 8 and no application has been added. Next, the system administrator can click "Add" button to add this new application to the system. The system stores all application programs in directory mdms\progs and loads currently used applications into system application library. After the administrator click "Add" button, system searches the program name in this directory. If it can not be found, the system displays a message box to indicate that the application program doesn't exist. Otherwise, the system loads the application program into application library well updates the system information on administrator interface window. Figure 12 shows the result, the number of applications has been changed from 8 to 9 and the application added is "Inventory Planning". If the administrator doesn't want an application he just added previously, he can click "Undo" button to delete it. MDMS stores application information in system file "apps". This file holds records of all the applications currently in application library. Each record contains the caption, program name, owner's login name and category of an application. Before applications are displayed on window, the data held in this file is assigned into an array of structure. This array then provides information needed to create the window. For example, we can find all the applications owned by a particular user who has same name with the current user by looping through this array. The total number and captions of these applications can be used to create the current user view. And the categories of these applications can be used as filters to change the appearance of user view. Table 2 shows the contents of this file.

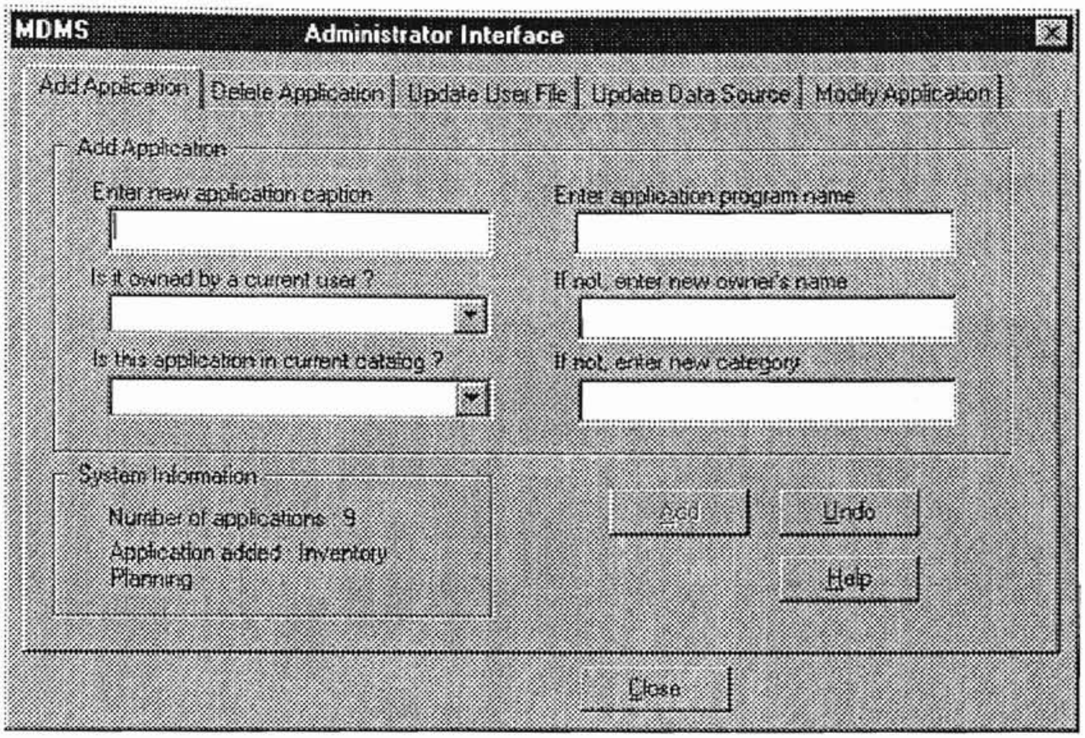


Figure 12. After Adding in Application "Inventory Planning"

Caption	Program	Owner	Category
Air pollution data analysis	air_polut	Environment	SMITH
Air temperature data analysis	air_temp	Weather	SMITH
Annual rainfall report	annu_rain	Weather	ANNA
Global weather forecast	glob_weather	Weather	ANNA
Inventory planning	invent_plan	Production	SMITH
Production planning	prod_plan	Production	ANNA
Productivity analysis	prod_analy	Production	ANNA
Regional sales analysis	sale_analy	Production	SMITH
Regional soil sample analysis	soil_analy	Environment	SMITH
Regional weather forecast	reg_weather	Weather	SMITH
Supply and demand analysis	supp_demand	Production	ANNA
Temperature and water supply	temp_water	Environment	SMITH
Water resource data analysis	water_resource	Environment	ANNA
Water supply and demand forecast	water_SandD	Environment	ANNA

Table 2. System Application File

Delete Application

Figure 13 shows the second page of the property sheet. This page supports application deletion task. It contains an edit box that allows the administrator to enter

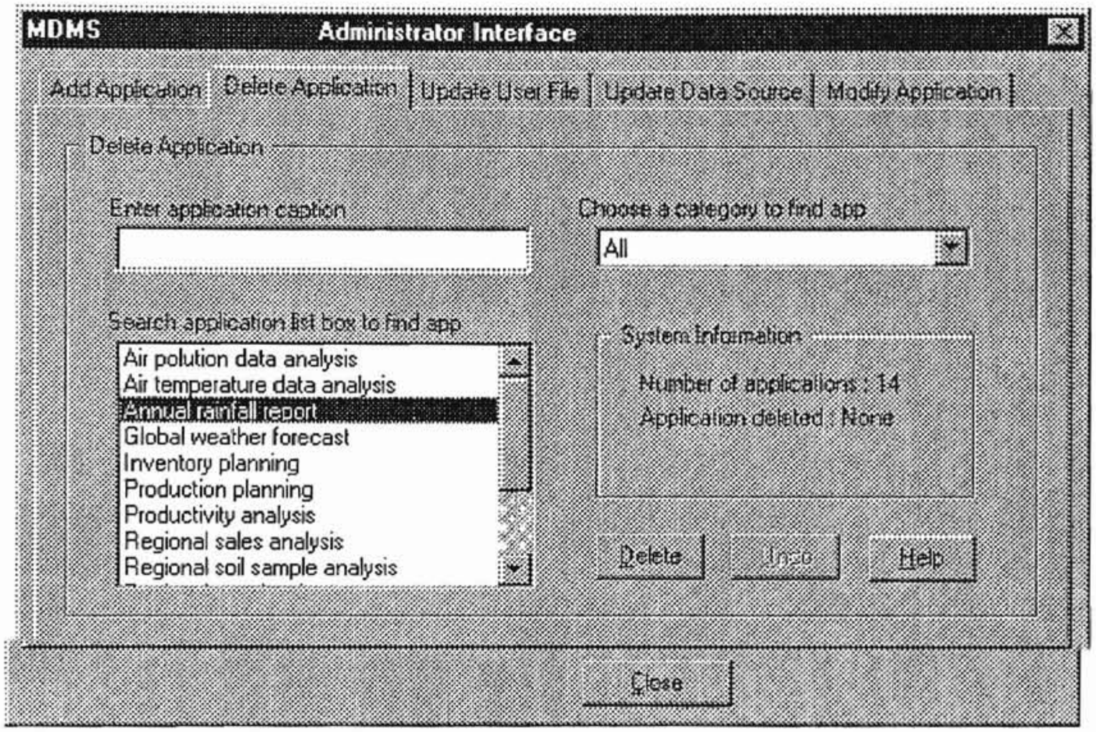


Figure 13. Application Deleting Page

application caption. Applications are displayed on a list box and a combo box of application category serves as a filter to change the appearance of the applications in the list box. The way to search an application is the same way used in the user interface window introduced in section 5.3. The administrator can either enter an application caption or choose one from the list box to delete the application. Figure 13 shows the window before deletion. The system information indicates that currently the application library has 14 applications and no application has been deleted. Figure 14 illustrates the result after the administrator deletes application "Inventory Planning". If the administrator accidentally deleted an application that should not be deleted. He can uses "Undo" button to bring the application back.

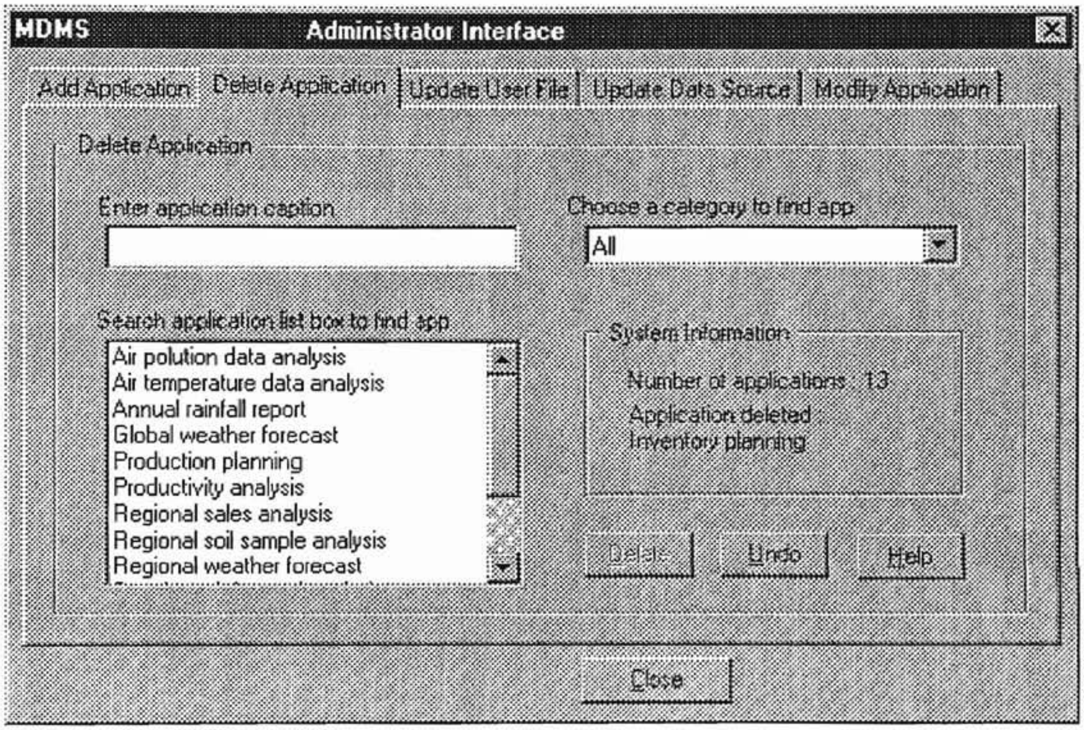


Figure 14. After Deleting Application "Inventory Planning"

Update User File

MDMS has a user register file that contains all the users' login name and password. When a new user is authorized to use the system or a current user becomes no longer eligible, the user file has to be modified. Figure 15 introduces the update user file page of administrator property sheet. This page contains two edit boxes used to enter new user's login name and password. After entering these data, the administrator can click "Add" button to add the new user. The login names and passwords of current users are displayed on a list box. To delete a current user, the administrator can either highlight

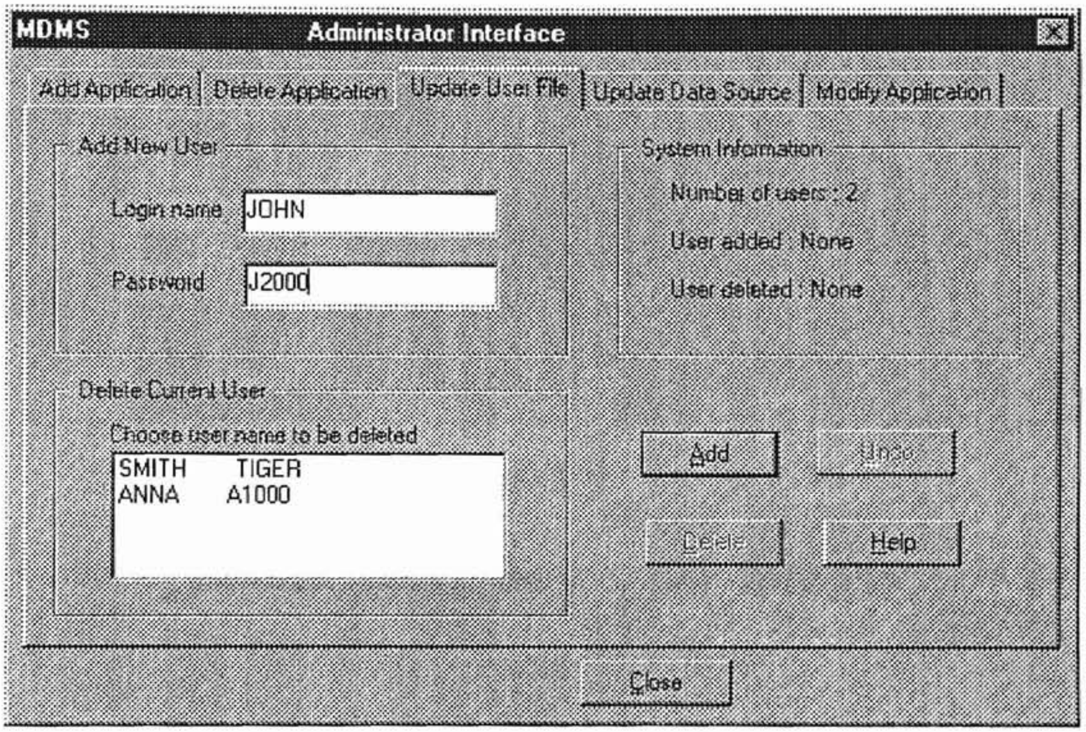


Figure 15. User File Updating Page

the user's login name and password then click "Delete" button or just double click on that login name and password. No matter what he did wrong, the administrator always can use "Undo" button to bring the previous status back. Figure 15 shows that there are only two users Smith and Anna and none of addition or deletion has been done. Figure 16 shows the result of adding new users John and Mark, the total number of users is four. The

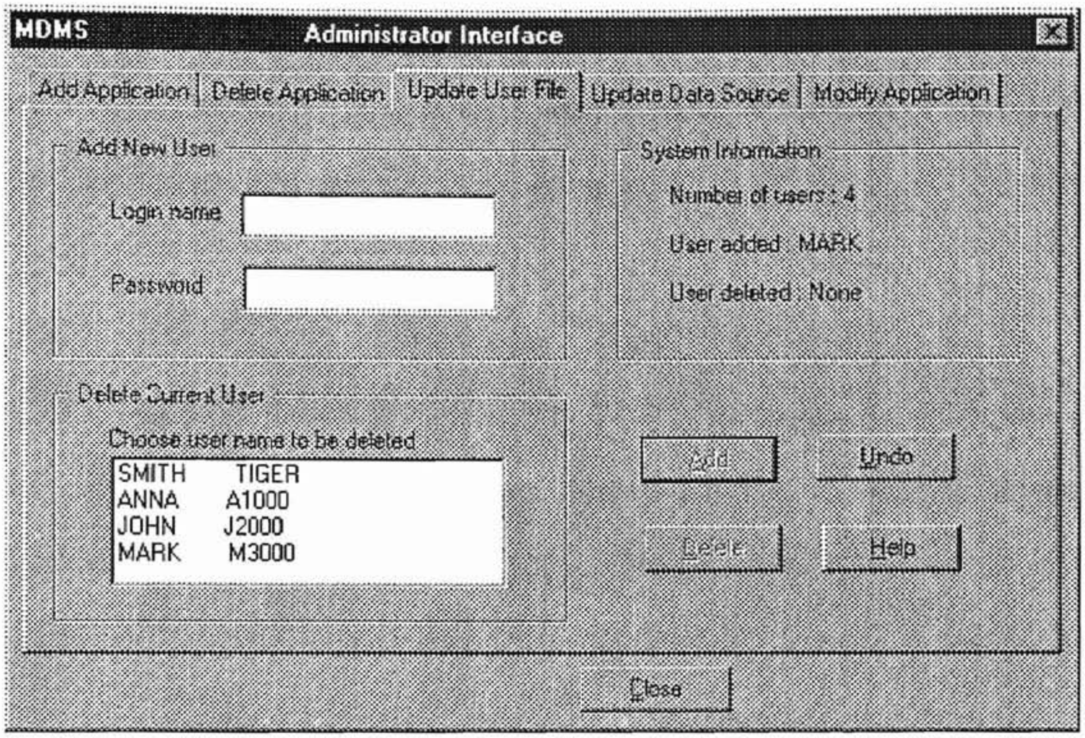


Figure 16. After Adding in Users "John" and "Mark"

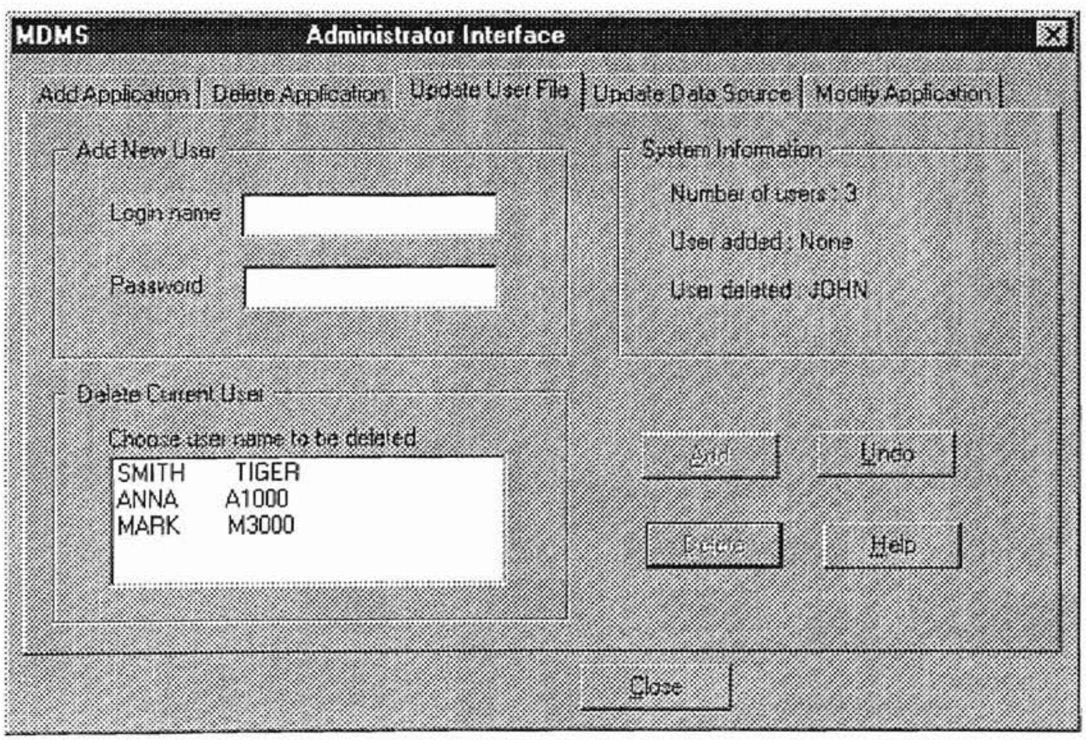


Figure 17. After Deleting User "John"

previously added user is Mark and no user has been deleted. Figure 17 is the result after John is deleted. The number of users is decreased to three. When each of these changes happens, the contents of system user file is also updated.

Update Data Source

The system administrator interacts with ODBC data sources through update data source task page. When it opens, this page displays information about the version of currently installed ODBC package, ODBC drivers and data sources. Figure 18 shows this page. After the administrator click "Open" button, the ODBC data source window shows up. This window displays all currently used ODBC data sources. Some of them are

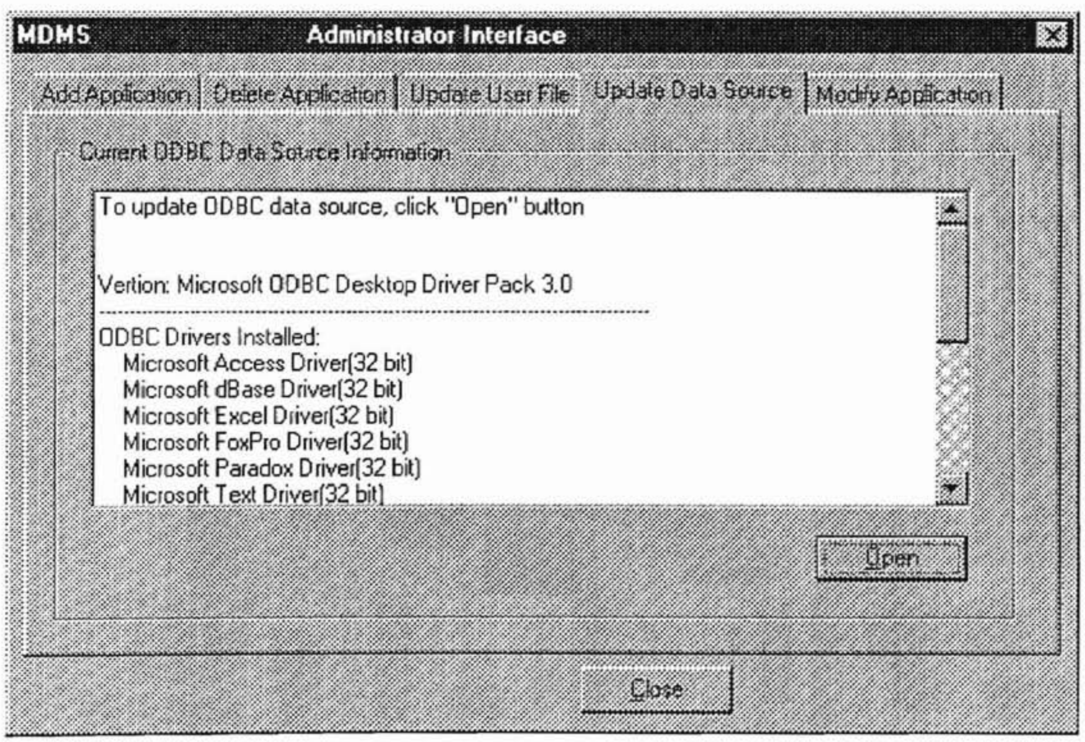


Figure 18. Data Source Updating Page

Microsoft default data sources. Figure 19 shows the ODBC data source window. To delete a data source, the administrator can highlight that data source then click "Delete" button to delete it. To add a new data source, he can click "Add" button and this will open

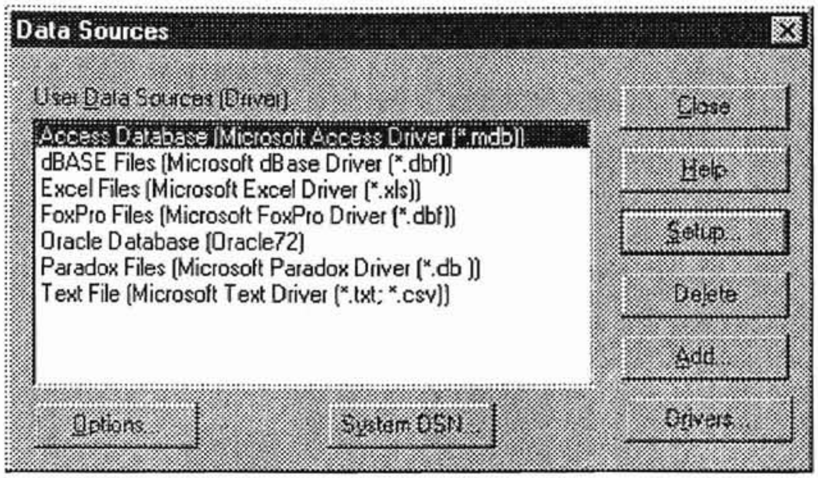


Figure 19. ODBC Data Source Window

an "Add Data Source" window. This window is shown by Figure 20. It displays all currently installed ODBC data sources. The administrator can highlight the one he needs

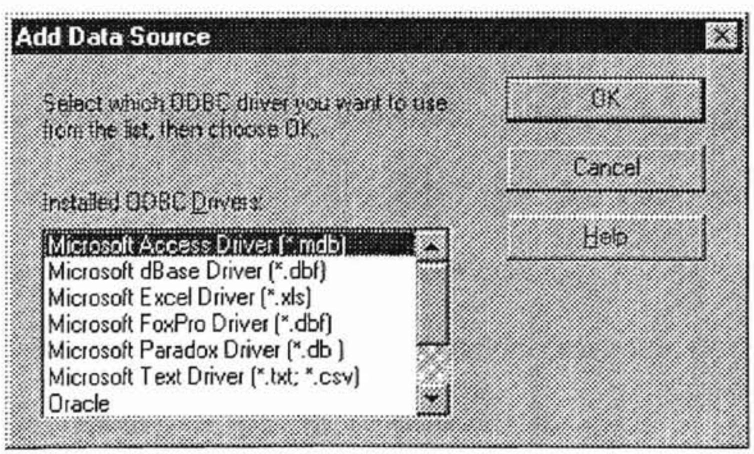


Figure 20. Add Data Source Window

then click "OK" button to add it. After this, a "Set up" window shows up. Using this window, the administrator can set up a particular data source he needs. Figure 21 shows the set up of an Access 7.0 database data source.

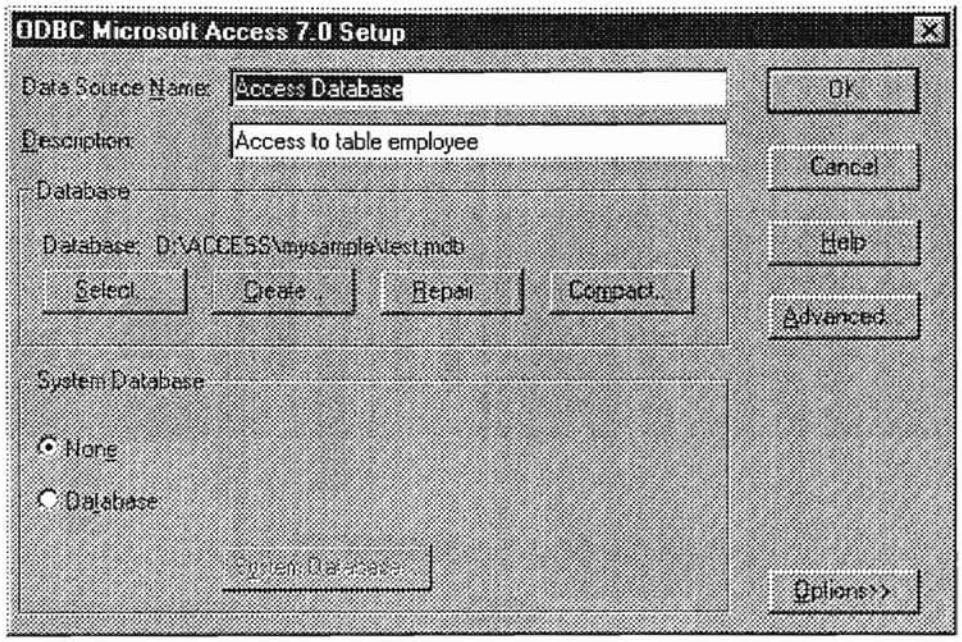


Figure 21. Access 7.0 Data Source Set Up

Modify Application File

In section 5.4.1, the functionality and updating of system application file "apps" is introduced. This section illustrates how to modify "apps" file. Each record in file "apps" holds information of a particular application. These information include caption, program name, owner's login name and category. It is possible that some of these information have to be changed due to some reasons. For instance, the owner of an application may change from Smith to John. An application caption can be replaced by a new one that is

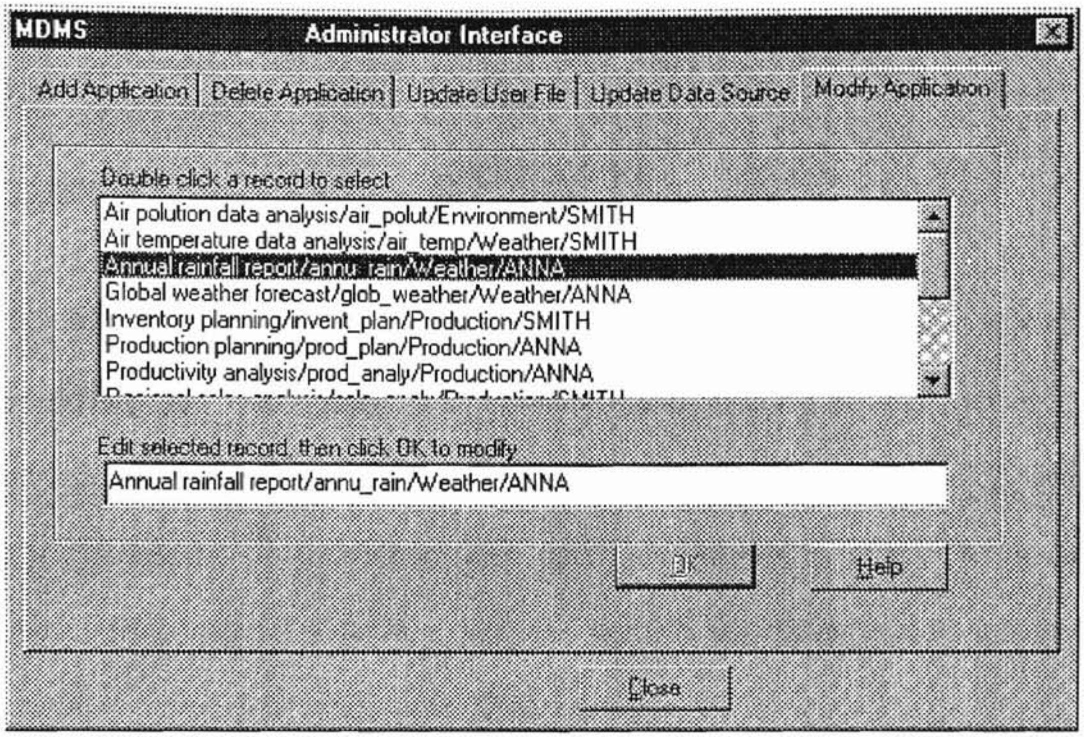


Figure 22. Application File Modifying Page

more meaningful. When these happen, the administrator needs a way to interact with “apps”. This way is provided by “Modify Application File” task page. Figure 22 shows this page. On this window, all the application records are displayed in a list box, each field in a record is separated from the others by “\”. To modify an application record, the administrator can click on that entry. This will copy the record into the edit box where he can make any changes. After it is done, the administrator can click “OK” button to save his modification.

CHAPTER VI

APPLICATION

To test MDMS system functionality, a multiple database application "Temperature and water supply" is developed and installed in the system. This application accesses two heterogeneous databases, one is Microsoft Access 7.0 personal database containing air temperature data tables and another is Oracle 7.2 personal database storing tables of water flow data. The application retrieves data from both data sources then creates result tables on the Oracle database.

Data Sources

The air temperature data tables are created based on the daily air temperature observation data contained in the Record of Evaporation and Climatological Observation. This data records the daily air temperature in Stillwater area during the period from January 1993 to October 1996. Four tables are created using this data, each of them holds daily average air temperature of a particular year in this period. These tables are stored in Microsoft Access 7.0 personal database as table airtemp93, airtemp94, airtemp95 and airtemp96 (Appendix: Air temperature data tables).

The water flow data tables are created based on the data provided by Monthly Operation Report of Stillwater Water Treatment Plants. This data includes daily water flow produced by the plants from January 1993 to October 1996. This data produces four tables, each of them records the daily water flow during a particular year from 1993 to

1996. These tables are stored in Oracle 7.2 personal database as table water93, water94, water95 and water96 (Appendix: Water flow data tables).

Tables and Data Calculation

There are many factors that influence the planning of water production. One of them is air temperature. The goal of this application is to manipulate data retrieved from multiple data sources and provide data analysis support for such planning. The application retrieves air temperature and water flow information from Access and Oracle databases and generates three result tables on the Oracle database. The first table contains monthly average air temperature for each year from 1993 to 1996. The second contains monthly average water flow for each year of the same period. The third summarize monthly average air temperature, monthly average water flow, monthly air temperature variation and monthly water flow variation. The following formulas are used to calculate data in these tables.

1. Formula used to calculate monthly average air temperature of each year:

$$\overline{T}_j = 1/n_j \sum_{i=1}^{n_j} t_{ij} \qquad (1)$$

j: Month, $j = 1, 2, \ldots\ldots, 12$.

n_j: Total days of month j, $28 \leq n_j \leq 31$.

i: Day, $i = 1, 2, \ldots\ldots, n_j$.

$\overline{T}_j$: Average air temperature of month j (in °F).

t_{ij}: Daily average air temperature of month j on day i(in °F).

2. Formula used to calculate monthly average water flow of each year:

$$\overline{W}_j = 1/n_j \sum_{i=1}^{n_j} w_{ij} \qquad (2)$$

$\overline{W}_j$: Average water flow of month j (in million gallons).

w_{ij}: Daily average water flow of month j (in million gallons).

3. Formula used to calculate monthly average air temperature of the period (Jan. 1993 - Oct. 1996):

$$\overline{T}_{oj} = 1/N \sum_{k=1993}^{1996} [\sum_{i=1}^{n_{kj}} t_{ij}], \text{ where } N = \sum_{k=1993}^{1996} n_{kj} \qquad (3)$$

$\overline{T}_{oj}$: Overall average air temperature of month j in the period (in °F).

t_{ij}: Daily average air temperature of month j on day i(in °F).

n_{kj}: Total days of month j in year k, $113 \leq n_{kj} \leq 124$.

N: Total days of month j in the period.

4. Formula used to calculate monthly average water flow of the period (Jan.1993 - Oct. 1996):

$$\overline{W}_{oj} = 1/N \sum_{k=1993}^{1996} [\sum_{i=1}^{n_{kj}} w_{ij}] \qquad (4)$$

$\overline{W}_{oj}$: Overall average water flow of month j in the period (in million gallons).

w_{ij}: Daily average water flow of month j (in million gallons).

5. Formula used to calculate the average air temperature in the period (Jan.1993 - Oct.1996):

$$\overline{T} = 1/12 \sum_{j=1}^{12} \overline{T}_{oj} \qquad (5)$$

$\overline{T}$: Average air temperature in the period (in °F).

6. Formula used to calculate the average water flow in the period (Jan.1993 - Oct.1996):

$$\overline{W} = 1/12 \sum_{j=1}^{12} \overline{W}_{oj} \tag{6}$$

$\overline{W}$: Average water flow in the period (in million gallons).

7. Formula used to calculate monthly air temperature variation:

$$V_{tj} = \overline{T}_{oj} - \overline{T} \tag{7}$$

V_{tj}: Air temperature variation of month j (in °F).

8. Formula used to calculate monthly water flow variation:

$$V_{wj} = \overline{W}_{oj} - \overline{W} \tag{8}$$

V_{wj}: Water flow variation of month j (in million gallons).

Result Tables

Figure 23 shows table "temp_year_month", this table contains the monthly air temperature of each year in the period from January 1993 to October 1996. This result average is generated based on the data retrieved from Access 7.0 database, the formula used to calculate this result is formula (1) introduced in section 6.2. Each data value in this table corresponds to $\overline{T}_j$ in formula (1). 0 values appear on column "year1996" and row "November" and "December" indicating that data for these two months are not available. The reason is that the observation period of the original record is from January

1993 to October 1996.

Personal Oracle7 Navigator - [TEMP_YEAR_MONTH]

	MONTH	YEAR1993	YEAR1994	YEAR1995	YEAR1996
1	January	35.29	34.94	38.13	33.81
2	February	38.86	36.14	44.04	40.72
3	March	47.45	51.35	50.26	44.52
4	April	56.23	58.6	57.13	57.43
5	May	65.97	66.74	63.94	73.39
6	June	76.17	80.2	74	77.73
7	July	83.35	80.03	81.48	81.13
8	August	82.58	79.26	81.68	78.32
9	September	67.9	70.07	69.63	69.2
10	October	56.61	62.06	61.42	60.52
11	November	43.2	50.23	51.87	0
12	December	41.71	41.81	39.42	0

For Help, press F1

Figure 23. Stillwater Monthly Average Air Temperature (in °F)

Figure 24 shows table "water_year_month", this table contains the monthly average water flow of each year in the same period of the previous table. This result is generated based on the data retrieved from tables "water93", "water94", "water95" and "water96" stored in Oracle 7.2 database. The formula used to calculate this result is formula (2) introduced in section 6.2. Each data value in table corresponds to $\overline{W}_j$ in formula (2). To make air temperature data and water flow data comparable, a same time period has been used to collect water flow data, that is, from January 1993 to October 1996. Due to this season, 0 values appear on column "year1996" and row "November" is also indicate that data for these two months are not available.

	MONTH	YEAR1993	YEAR1994	YEAR1995	YEAR1996
1	January	7.01	6.23	6.42	6.56
2	February	6.57	6.52	6.9	7.31
3	March	6.75	6.51	6.9	6.92
4	April	6.16	5.99	6.97	7.07
5	May	5.65	6.02	5.66	7.57
6	June	6.21	7.42	5.64	7.76
7	July	7.67	7.73	8.33	7.71
8	August	7.96	8.05	7.46	7.18
9	September	6.64	7.4	7.38	7.56
10	October	6.43	7.95	6.43	6.88
11	November	5.64	7.45	5.96	0
12	December	5.27	6.39	5.74	0

Figure 24. Stillwater Water Treatment Plants Monthly Average Water Flow (in Million Gallons)

Figure 25 shows table "temp_water", this table is a conjunct result of both air temperature data and water flow data. The second column "AVG_TEMP" means overall monthly average air temperature of the period from Jan.1993 to Oct.1996. The data values in this column is calculated by formula (3), and each value corresponds to $\overline{T}_{oj}$ in formula (3). The third column "AVG_WATER" means overall monthly average water flow of the same period. The data values in this column is calculated by formula (4), and each value corresponds to $\overline{W}_{oj}$ in formula (4). The last row of this table is "Average", this row contains the average air temperature and the average water flow of same period. The former is calculated by formula (5), the value is 58.87(°F) corresponding to $\overline{T}$ in formula (5). The latter is calculated by formula (6), the value is 6.8(million gallons) corresponding

Personal Oracle7 Navigator - [TEMP_WATER]

	MONTH	AVG_TEMP	AVG_WATER	TEMP_VAR	WATER_VAR
1	January	35.54	6.55	-23.33	-.24
2	February	39.95	6.83	-18.93	.03
3	March	48.4	6.77	-10.48	-.03
4	April	57.35	6.57	-1.52	-.23
5	May	67.51	6.22	8.63	-.57
6	June	77.03	6.78	18.15	-.02
7	July	81.5	7.86	22.63	1.06
8	August	80.46	7.65	21.59	.85
9	September	69.2	7.25	10.33	.45
10	October	60.15	6.92	1.28	.13
11	November	48.43	6.35	-10.44	-.45
12	December	40.98	5.8	-17.9	-.99
13	Average	58.87	6.8	0	0

For Help, press F1

Figure 25. Monthly Variation of Air Temperature (in °F)
and Water Flow (in Million Gallons)
(Jan. 1993 - Oct. 1996)

to $\overline{W}$ in formula (6). The 0 values in the last row of column "TEMP_VAR" and "WATER_VAR" indicate that the values of "Average" are not applicable in these columns. The fourth column "TEMP_VAR" means monthly air temperature variation of the same period. The data values in this column is calculated by formula (7), and each value corresponds to V_{tj} in formula (7). The fifth column "WATER_VAR" means monthly water flow variation of the same period. The data values in this column is calculated by formula (8), and each value corresponds to V_{wj} in formula (8).

CHAPTER VII

CONCLUSION

MDMS establishes a multiple database management system that integrates multiple database application support with system management functionality. This chapter summerize the advantages of MDMS and its possible future extension.

Summary of MDMS

Most of the advantages of MDMS come from its object oriented design. The system objects are implemented by classes, most of theses classes are derived from MFC base classes. The CDatabase derived class makes it easy to declare a database object in application then use it to obtain ODBC support. The CRecordSet derived class presents and handles the data selected from the target data sources. It is these inheritances that makes it possible to build up the multiple database support feature of MDMS.

Graphical user interface provides a flexible and efficient interaction between end user and program. Both the user and administrator interface windows support multiple ways of displaying, searching and executing. Whenever a system update takes place, the corresponding window message report the change immediately. This feature creates a user friendly system environment.

In object oriented programming, encapsulation combines data structures and functionality into objects. Communication between objects relies only on message passing. This feature provides the advantage to maintain the integrity of each part of the system. It also makes it possible to add a new system component without a successive modification over the others. This advantage will benefit the future system extension or development.

Future Extension of MDMS

Because MDMS's design and implementation adopt object oriented methodology, both system architecture and functionality can be extend or modified without triggering a chain reaction of program modification. To focus on the issue of multiple database accessing, the current version of MDMS implements some system functionality only at a basic level. For example, the current version of MDMS assumes that an application has only one owner, and it is not always the case. This kind of simplification leave a room for future modifications.

This discussion focus on the possible future extension of MDMS. One possible system extension is to add a "knowledge" library. A high level decision support application may involves many complex calculation, analysis and reasoning[4]. The programs or software designed to implement these methods may already big enough such that it is not practicable to include them into application program. Furthermore, some of these tools are needed to many applications. It will be redundant to make a copy of same tool for each application. The solution is to have a knowledge library that separates these tools from applications well provides a common interface between application and the library. To use an algorithm or function routine in this library, an application program simply declare an object of the class of this library, then plays a function call to desired algorithm or function routine. Figure 26 is a diagram to illustrate this extension.

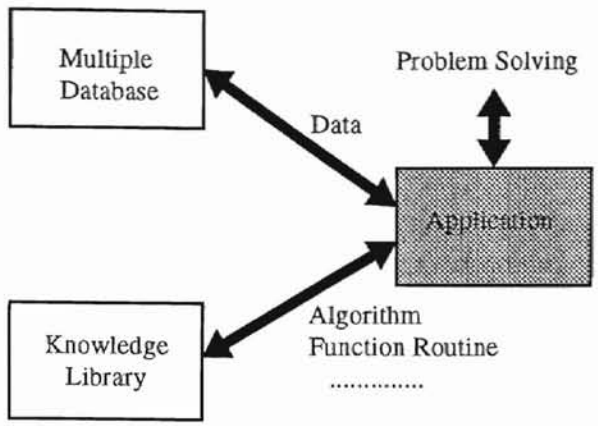


Figure 26. An Extension of A Knowledge Library

BIBLIOGRAPHY

Barsalou, T., Sujansky, W., Herzenberg, L. A., & Wiederhold, G. (1991). Management of Complex Immunogenetics Information Using an Enhanced relational Model. Computer and Biomedical Research, 24(5), 476-498.

Subrahmanian, V. S., Adali, S., Brink, A., Lu, J. J., Rajput, A., Rogers, T. J., Ross, R., & Ward, C. (1996). HERMES: A Heterogeneous Reasoning and Mediator System. The Department of Computer Science, University of Maryland.

Wiederhold, G. (1995). Mediation in Information System. ACM Computer Surveys, 27 (2), 265 - 267.

Sciore, E., Siegel, M., & Rosenthal, A. (1994). Using Semantic Values to Facilitate Interoperability Among Heterogeneous Information System. ACM Transaction on Database Systems, 19(2), 254 - 290.

Litwin, W., Risch, T. (1994). Applications of Databases. First International Conference, ADB - 94, Sweden.

Reddy, M. P., Siegel, M.& Gapta, A. (1992). Towards an Active Schema Integration Architecture for Heterogeneous Database System. IEEE International Workshop on Research Issues on Data Engineering, Interoperability in Multidatabase Systems. Vienna, Austria (pp.178-183).

Arens, Y., Chee, C. Y., Hsu, C. H. & Knoblock, C. (1993). Retrieving and Integrating Data From Multiple Information Sources. International Journal of Intelligent Cooperative Information Systems. 2(2), 27-158.

Bright, M. W., Hurson, A. R. & Pakzad, S. (1994). Automated Resolution of Semantic Heterogeneity in Multi - databases. ACM Transaction on Database Systems. 19 (2), 212-253.

Sciore, E., Siegel, M. & Rosenthal, A. (1994). Using Semantic Values to Facilitate Interoperability Among Heterogeneous Information Systems. ACM Transaction on Database Systems. 19(2), 254-290.

Ishikawa, H., Suzuki, F. (1993). The Model, Language, and Implementation of an Object-Oriented Multimedia Knowledge Base Management System. ACM Transaction on Database Systems. 18(1), 1 - 50.

APPENDIX

APPLICATION DATA SOURCES

This appendix presents data sources of the application introduced in Chapter VI. The data sources consist of the air temperature data tables in Microsoft Access 7.0 database and the water flow data tables in Oracle 7.2 database. To display the database environment together with these data tables, print screen is used to store each table into clipboard. Due to the limitation of computer screen, only part of the data contained in each table can be displayed. In fact, each table contains dialy data of each month for a particular year.

Air Temperature Data Tables

The air temperature data tables are created based on the daily air temperature observation data contained in the Record of Evaporation and Climatological Observation. This data records the daily air temperature in Stillwater area during the period from January 1993 to October 1996. Four tables are created using this data, each of them holds daily average air temperature of a particular year in this period. These tables are stored in Microsoft Access 7.0 personal database as table airtemp93, airtemp94, airtemp95 and airtemp96.

Microsoft Access - [Table: airtemp93]

File Edit View Format Records Window Help

Day	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1	22.5	47	43.5	46.5	64	68.5	85	92	66.5	74	33.5	47.5
2	30	45.5	43.5	36.5	67	71.5	82.5	86.5	73	72	50	48.5
3	37.5	48	41.5	48	56	74.5	84	81	76	57	43.5	46.5
4	46	49.5	42.5	52	57.5	73.5	84.5	80	66	59	50.5	53
5	30	49	40.5	45	66.5	64.5	81.5	74.5	70.5	68	55	47.5
6	33	38.5	43	46	71	68.5	83	66.5	73	68	34.5	47.5
7	36.5	45	46.5	53.5	73	81.5	78	68	70.5	72.5	37	41
8	40	49	52.5	58	69.5	75	80	73	70.5	73.5	45	41.5
9	36	55	58	45.5	64	74	83	80	62.5	62.5	47.5	43.5
10	38	52	52.5	58.5	58	73.5	81	84.5	71.5	48.5	42.5	52.5
11	28	47	46.5	62.5	56	73	81.5	86	70.5	47	47.5	46.5
12	35	38	38	69	57.5	73.5	81.5	87	75	54	58.5	50
13	28.5	36	33	67	64.5	77	81	85.5	80	59.5	58	55
14	23.5	40	29.5	55	61	78	80	86.5	64.5	60.5	48.5	40.5
15	24.5	38	46	54	66.5	77	77	88	51.5	61	48.5	35
16	38	23.5	50	48	75.5	81	80.5	87.5	54	60.5	39	36.5
17	39.5	24	48.5	56.5	71.5	81.5	82.5	87	58.5	65	38	40.5
18	34.5	12	39	52	72.5	79.5	84.5	85.5	65.5	58	41.5	41
19	26.5	21.5	39.5	64	61	78	84	85.5	70	60.5	50.5	38.5

Record: 1 of 31

Datasheet View NUM

Stillwater Daily Air Temperature in 1993 (°F)

Microsoft Access - [Table: airtemp94]

File Edit View Format Records Window Help

Day	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1	49	22	42	54.5	45	79	82.5	78	68.5	73.5	48.5	42
2	43	29	32.5	67.5	53.5	77.5	82	82.5	66	72.5	57	51.5
3	39.5	32.5	43	57	48.5	78	83	78	72	72.5	65.5	55
4	31	38	54	54	59	75.5	86	81.5	74	70.5	66	58.5
5	32.5	44	58	57	58.5	76	85.5	81.5	81	74	51.5	52.5
6	48.5	44	60	33.5	68.5	77	86.5	72	75	73	44.5	38.5
7	24.5	45	60	40.5	69.5	80.5	82.5	76.5	68	75	55.5	39.5
8	19.5	30	45.5	49.5	52.5	83	81	75	71.5	64	60.5	34
9	34	15.5	36.5	62.5	59	79	76	77.5	72.5	55.5	62	31
10	40.5	15.5	36.5	63.5	66	72	75.5	78	72	56	43.5	36
11	41	24.5	40	55.5	69	80	81	82	73	53.5	48.5	31
12	32.5	36.5	52.5	53.5	72	75	80	83.5	74.5	54	52	33
13	42	31.5	52.5	49.5	74	81.5	80	83.5	78	56	52.5	35.5
14	41	37	54	63.5	71.5	82.5	81	85.5	80.5	61.5	59	36.5
15	30.5	41.5	62	71	66	82	78	68	81.5	59	48.5	41
16	27.5	42.5	50	53	67.5	79	75.5	70	71	61	42.5	40.5
17	24	46.5	56.5	62	75.5	80.5	80	74	68	67	46	42
18	16	59	67	65.5	70	79.5	81.5	78.5	67	67.5	44	45

Record: 6 of 31

Datasheet View NUM

Stillwater Daily Air Temperature in 1994 (°F)

Microsoft Access - [Table: airtemp95]

File Edit View Format Records Window Help

Day	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1	33.5	49.5	31	49.5	55.5	70	68	82	81.5	66	59	53.5
2	28.5	55.5	25	56	50	72.5	71	76	82.5	66	59	54.5
3	35.5	59.5	24.5	66.5	52	77.5	73.5	78.5	84.5	58	34	56.5
4	20	40.5	31	56	55	72.5	82	78.5	84	61	36.5	47
5	20.5	45.5	38	52.5	57.5	71	77.5	79.5	85.5	64	40.5	47.5
6	30.5	41.5	43.5	57.5	59	71	77.5	81.5	82	60.5	52	41
7	34	46.5	39.5	63	65	74.5	83	82.5	81.5	56.5	57	38.5
8	34	27.5	22.5	69	65	84.5	85	84	71.5	61	47.5	34
9	33.5	35	29	72.5	64	79	86.5	84.5	63.5	64	44.5	29
10	40.5	44	47	59.5	64.5	70.5	83.5	80	70.5	65.5	63.5	19
11	43	35	53.5	40.5	56.5	64	84.5	82	72	70.5	50	30
12	52.5	29	58	43.5	63.5	65	87	83	70	73	34	40
13	47.5	24.5	61.5	51	67	68.5	87.5	81	75.5	73	54.5	44.5
14	41	33	58.5	56	75.5	73.5	86.5	84.5	78.5	57	45	49.5
15	39.5	43.5	61.5	68	69	75.5	84	84	79	56.5	45.5	49.5
16	45	34	59.5	65.5	74.5	75.5	82	83.5	70.5	65.5	42	48.5
17	50	34	61	68	75	77	82.5	83	74	66	47.5	49
18	44.5	38.5	61.5	60.5	68.5	73	83.5	83	76	70	183	45
19	37	43.5	64	54.5	55.5	75	82.5	82	74	72	49.5	34.5
20	39	48.5	70	54	64	76.5	81	85	67.5	61	58.5	34

Record: 11 of 31

Datasheet View NUM

Stillwater Daily Air Temperature in 1995 (°F)

Microsoft Access - [Table: airtemp96]

File Edit View Format Records Window Help

Day	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1	43.5	13.5	24	46	59.5	70.5	86.5	79	76.5	66	0	0
2	38	16	41	53	64.5	66.5	86.5	76.5	74.5	66	0	0
3	29.5	13	47.5	67	69.5	72.5	88.5	79.5	77	63.5	0	0
4	33.5	-2	47	56.5	75.5	72	89	79.5	75.5	61.5	0	0
5	34	21	57	45.5	81	74.5	81.5	83.5	77	62.5	0	0
6	26	27.5	46	37	77	78.5	92	84.5	78.5	61.5	0	0
7	18.5	33.5	23	48.5	78	75	97	84	77	67.5	0	0
8	18.5	50.5	21	55.5	78	66.5	87	82.5	75.5	64	0	0
9	36	48.5	20	52.5	78.5	65	80	81	71.5	58.5	0	0
10	42.5	57.5	34.5	58.5	77	71.5	67	79	75	61.5	0	0
11	42.5	48.5	49.5	63.5	65	74.5	66.5	73.5	77.5	56.5	0	0
12	46	39	54.5	69.5	58.5	79	67.5	72.5	80	61.5	0	0
13	46	42	55	50.5	64.5	77.5	76.5	73	72.5	69	0	0
14	51	51.5	65	56.5	58.5	81	78.5	75	70	71	0	0
15	48.5	53	58.5	51.5	73.5	79	75	77.5	66.5	69	0	0
16	39.5	36	49	50	80	75.5	79	79.5	65.5	70	0	0
17	51.5	29.5	53	63	82.5	83	83.5	79.5	66	70.5	0	0
18	35.5	47	54	74.5	82	74	86.5	77.5	68.5	53	0	0

Record: 12 of 31

Datasheet View NUM

Stillwater Daily Air Temperature from Jan. to Oct. 1996 (°F)

Water Flow Data Tables

The water flow data tables are created based on the data provided by Monthly Operation Report of Stillwater Water Treatment Plants. This data includes daily water flow produced by the plants from January 1993 to October 1996. This data produces four tables, each of them records the daily water flow during a particular year from 1993 to 1996. These tables are stored in Oracle 7.2 personal database as table water93, water94, water95 and water96.

Personal Oracle7 Navigator - [WATER93]

File Edit View Window Help

	DAY	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
1	1	5.612	6.741	5.668	7.629	4.578	5.973	7.245	9.308	5.887	6.401	6.918	5.11
2	2	5.81	6.725	7.066	6.298	7.882	6.045	8.248	8.98	7.326	8.202	4.379	4.692
3	3	5.628	6.709	7.189	6.285	5.626	5.864	8.443	7.084	7.132	7.715	4.833	4.225
4	4	6.712	7.387	7.91	6.543	5.522	5.78	7.474	3.047	6.842	6.426	5.834	5.816
5	5	6.635	6.059	5.724	4.512	4.74	6.893	7.136	0	7.659	6.122	5.393	7.037
6	6	6.768	6.135	8.145	6.725	4.773	6.874	7.844	0	6.167	6.405	6.578	5.136
7	7	6.657	5.891	7.803	6.756	5.31	5.304	5.774	0	6.909	7.224	7.215	4.975
8	8	6.476	6.745	7.873	6.652	7.586	5.887	7.346	0	6.522	5.846	6.742	4.278
9	9	6.82	7.228	7.658	7.488	5.474	5.787	4.903	0	5.762	6.654	5.317	5.381
10	10	6.785	7.105	6.886	8.612	5.017	5.036	8.285	1.493	7.123	7.208	4.967	4.977
11	11	6.672	6.02	6.013	6.868	6.179	5.517	11.026	5.083	7.995	7.223	4.934	6.273
12	12	6.674	6.521	5.225	5.805	7.017	5.741	5.862	6.8	7.561	4.119	5.753	6.301
13	13	6.881	5.614	4.768	7.664	5.743	7.828	5.618	6.002	8.402	5.86	5.788	4.918
14	14	7.473	5.901	4.969	7.714	5.485	6.856	4.749	7.229	4.955	6.186	6.022	4.102
15	15	7.667	5.487	7.785	6.266	7.149	5.322	4.733	10.07	5.116	5.38	7.173	4.77
16	16	9.16	6.721	8.094	5.065	6.654	6.087	6.025	9.315	5.766	8.296	5.606	4.804
17	17	6.901	7.633	6.734	5.634	5.364	6.325	7.723	8.043	6.679	7.59	4.728	4.814
18	18	7.397	6.686	6.72	6.008	4.881	6.822	8.107	7.754	7.688	7.756	6.428	5.238
19	19	7.913	7.556	6.634	6.286	4.05	7.295	8.351	9.522	7.528	6.378	5.471	5.73
20	20	7.626	6.4	5.587	5.05	4.365	5.679	6.29	9.515	6.126	5.224	5.816	5.835
21	21	7.592	5.836	5.891	4.754	3.829	6.405	7.517	10.597	4.82	3.712	6.934	5.349
22	22	7.772	6.682	5.876	4.88	6.794	6.06	8.246	9.192	5.143	5.313	5.93	4.853

Stillwater Water Treatment Plants Daily Water Flow in 1993 (Million Gallons)

Personal Oracle7 Navigator - [WATER94]

File Edit View Window Help

	DAY	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
1	1	5.277	6.809	6.792	8.541	6.525	7.21	8.199	7.235	6.834	7.444	7.425	7.65
2	2	5.478	5.762	6.787	6.171	6.268	5.544	9.454	6.834	6.138	9.155	8.926	7.216
3	3	6.298	6.083	6.889	5.833	5.206	6.126	9.635	6.546	6.34	9.409	8.692	6.793
4	4	6.187	6.888	7.08	7.133	4.807	7.601	10.18	6.678	7.676	9.219	7.57	6.687
5	5	4.659	6.093	6.937	5.785	4.992	7.618	9.331	6.93	8.031	6.293	7.114	7.7
6	6	6.36	7.257	5.153	4.848	5.858	7.5	8.73	8.199	8.423	7.654	7.769	7.196
7	7	6.109	6.349	6.283	6.403	6.012	5.713	9.113	7.566	6.371	7.767	8.625	6.578
8	8	5.761	6.266	5.626	6.262	6.352	5.352	7.99	6.821	5.767	10.795	7.913	5.918
9	9	6.258	6.306	4.826	6.638	5.669	5.755	10.537	7.215	6.572	9.429	7.506	7.734
10	10	6.561	6.929	7.035	6.821	4.644	5.939	11.175	7.904	8.158	8.45	6.914	7.384
11	11	6.752	6.548	6.438	5.793	4.918	5.764	9.184	8.162	7.434	8	7.403	6.761
12	12	6.812	6.3	5.898	4.708	5.758	7.873	8.837	8.916	6.649	8.21	8.653	6.3
13	13	6.001	6.778	5.437	5.951	5.938	5.599	7.8	11.925	6.747	8.157	7.244	6.658
14	14	5.839	6.852	6.862	6.24	5.857	6.317	6.938	9.531	6.77	8.444	7.258	7.364
15	15	6.698	7.193	7.265	6.312	7.543	6.287	5.992	7.636	6.719	6.983	7.906	6.787
16	16	6.533	6.465	6.211	6.628	6.146	5.916	7.262	8.73	6.66	5.84	8.474	6.398
17	17	6.3	6.3	6.299	7.798	5.322	7.726	6.91	9.981	6.728	7.736	7.605	5.841
18	18	6.486	6.811	6.409	6.857	5.736	8.246	6.885	6.713	6.917	7.896	7.553	5.779
19	19	6.373	7.071	6.361	3.067	6.154	8.721	6.831	8.405	7.908	8.788	7.748	5.858
20	20	6.552	6.154	7.014	0	6.793	7.169	6.816	8.517	6.152	7.432	7.648	6.779
21	21	6.313	5.286	6.184	0	7.593	7.292	6.146	8.46	6.518	6.7	8.522	6.708
22	22	6.434	7.629	6.817	0	7.686	7.997	5.232	5.776	6.599	7.421	8.349	5.788

Stillwater Water Treatment Plants Daily Water Flow in 1994 (Million Gallons)

Personal Oracle7 Navigator - [WATER95]

File Edit View Window Help

	DAY	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
1	1	5.205	6.847	8	8.553	6.612	4.881	6.336	5.187	8.995	9.127	5.821	6.014
2	2	5.646	5.803	6.237	6.709	5.476	5.692	7.409	4.889	8.272	5.859	5.649	7.274
3	3	6.631	6.011	8.025	6.039	5.377	4.821	7.622	4.816	10.982	4.803	5.634	6.808
4	4	6.786	5.747	7.935	6.572	5.687	6.509	7.305	5.134	11.803	5.287	5.743	6.09
5	5	6.725	6.721	8.313	6.692	5.914	5.5	8.242	5.673	10.021	5.722	7.421	6.07
6	6	5.755	6.805	7.192	6.657	5.756	3.812	8.337	5.666	8.607	5.685	5.992	5.546
7	7	6.394	6.762	5.081	8.333	5.8	1.429	8.63	5.899	7.298	6.78	5.824	5.147
8	8	7.69	6.64	5.963	7.238	5.675	0	8.064	7.046	5.831	6.691	5.671	5.771
9	9	6.813	6.106	7.62	7.776	5.471	0	10.273	5.646	7.696	6.676	5.799	6.194
10	10	5.791	6.425	5.692	7.603	4.81	1.45	8.829	5.911	8.605	6.276	6.334	6.68
11	11	6.63	6.8	5.341	7.24	5.041	2.924	8.5	6.798	7.782	5.721	5.072	6.423
12	12	7.731	6.101	6.327	6.008	5.682	3.76	9.618	7.883	7.437	5.887	6.066	4.741
13	13	7.435	6.436	6.662	7.731	5.848	4.356	10.857	8.629	5.974	6.09	7.257	6.8
14	14	5.709	6.766	6.441	8.284	5.89	5.495	10.443	7.843	6.364	6.5	5.489	5.704
15	15	6.231	6.503	5.604	7.075	5.789	5.436	11.158	6.681	6.632	8.509	5.742	5.47
16	16	6.758	5.896	8.158	6.858	5.762	5.11	10.874	6.771	6.816	7.528	6	5.188
17	17	7.262	8.178	7.646	6.73	5.7	8.193	7.87	6.687	6.666	5.901	5.707	6.162
18	18	7.271	8.827	6.881	6.165	5.383	8.345	11.024	7.603	7.633	5.697	7	6.09
19	19	5.893	6.4	6.774	6.386	4.445	7.633	8.038	8.63	6.351	5.736	7.201	5.032
20	20	6.033	6.834	6.574	6.505	5.32	6.609	10	10.437	5.703	6.892	6.043	5.901
21	21	6.673	7.3	8.132	6.761	7.693	5.816	5.838	9.537	6.48	6.34	4.989	6.715
22	22	5.81	8.23	8.118	6.712	6.898	5.866	6.595	7.335	5.831	7.75	6.306	5.336

Stillwater Water Treatment Plants Daily Water Flow in 1995 (Million Gallons)

Personal Oracle7 Navigator - [WATER96]

File Edit View Window Help

	DAY	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
1	1	5.771	6.966	6.85	6.948	7.554	6.472	10.116	5.795	7.42	6.65	0	0
2	2	5.825	7.892	7.996	7.085	7.922	5.804	8.835	5.844	9.022	6.649	0	0
3	3	5.768	7.996	7.944	7.285	7.494	7.352	8.845	5.865	8.353	6.733	0	0
4	4	5.808	7.738	6.492	5.976	9.503	6.546	12.088	6.2	6.825	7.291	0	0
5	5	5.794	8.317	6.75	8.659	9.537	6.831	9.722	7.152	7.43	10	0	0
6	6	5.701	8.544	6.719	6.137	7.928	6.857	11.254	6.494	7.986	10.846	0	0
7	7	6.977	7.57	4.254	6.709	7.71	6.575	12.739	7.737	10.128	7.075	0	0
8	8	7.015	7.667	7.163	7.606	7.846	6.146	9.038	6.005	10.478	5.998	0	0
9	9	8.267	6.429	6.259	8.212	8.38	5.832	6.702	7.555	8.445	7.474	0	0
10	10	7.146	7.501	6.112	8.888	7.631	7.986	5.247	7.789	8.193	7.143	0	0
11	11	6.598	7.028	7.49	7.499	5.985	7.37	5.548	6.742	9.831	6.142	0	0
12	12	6.447	7.081	7.699	7.942	6.283	7.954	5.89	5.84	9.17	5.983	0	0
13	13	6.136	7	7.723	7.946	6.035	7.745	6.307	5.774	8.803	7.735	0	0
14	14	5.71	7.566	8.5	8.076	5.659	8.835	6.585	5.704	8.608	8.398	0	0
15	15	6.854	7.715	7.949	5.316	6.238	7.861	6.799	7.306	6.336	7.642	0	0
16	16	6.573	6.853	6.982	8.147	5.741	7.538	6.712	7.992	6.336	7.276	0	0
17	17	6.757	6.279	7.654	7.49	7.332	6.652	5.854	7.406	6.104	6.476	0	0
18	18	6.792	6.715	7.298	7.788	8.05	6.658	6.972	7.718	6.597	6.217	0	0
19	19	7.069	8.143	6.334	7.649	9.682	6.633	9.146	7.583	6.713	8.344	0	0
20	20	6.583	7.765	6.668	7.201	8.327	8.365	9.6	7.714	6.743	8.215	0	0
21	21	7.837	7.702	7.043	7.2	8.209	8.961	9.52	8.715	6.787	6.864	0	0
22	22	6.48	7.232	6.859	6.394	7.759	8.918	10.259	8.634	7.166	6.626	0	0

Stillwater Water Treatment Plants Daily Water Flow
from Jan. to Oct. 1996 (Million Gallons)

VITA

Jinghua Liu

Candidate for the Degree of

Master of Science

Thesis: A MULTIPLE DATABASE MANAGEMENT SYSTEM: MDMS

Major Field: Computer Science

Biographical:

Education: Graduated from Columbia College, Columbia, Missouri in 1992; received Bachelor of Arts. Completed the requirements for the Master of Science degree with a major in Computer Science at Oklahoma State University in July 1997.

Experience: Graduate teaching assistant, Oklahoma State University, Department of Computer Science, January 1995 to May 1997; Engineer in Beijing Institute of Mechanical and Electrical Engineering from 1983 to 1988.

Professional Memberships: Member of Beijing Mechanical Engineering Association.