

RELATIONAL DATABASE AND OPTIMAL
HERBICIDE SELECTION IN CHEMICAL
WEED CONTROL

By

Zhuping Gao

Bachelor of Science
Shanxi Agricultural University
Shanxi, China
1982

Master of Science
Beijing Agricultural University
Beijing, China
1987

Master of Science
Kansas State University
Manhattan, Kansas
1997

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, 2000

RELATIONAL DATABASE AND OPTIMAL
HERBICIDE SELECTION IN CHEMICAL
WEED CONTROL

Thesis Approved:

A handwritten signature in black ink, appearing to read "D. E. Haden", written over a horizontal line.

Thesis Advisor

A handwritten signature in black ink, appearing to read "J. Chandler", written over a horizontal line.A handwritten signature in blue ink, appearing to read "Gary Beutling", written over a horizontal line.A handwritten signature in black ink, appearing to read "Alfred Sarlysi", written over a horizontal line.

Dean of the Graduate College

ACKNOWLEDGMENTS

I would like to take this opportunity to express my sincere gratitude and thanks to my major advisor, Dr. G. E. Hedrick, for his intelligent supervision, constructive guidance, inspiration and friendship. I wish to heartedly express my appreciation to Dr. J. P. Chandler and Dr. D. Heisterkamp for their assistance as committee members.

I would like to give my special appreciation to my wife, Lirong, for her precious suggestions to my research, her strong encouragement at times of difficulty, love, understanding and support throughout my studies and during the preparation of this thesis. I would also like to thank my daughter, Sally, my son, Richard for their love and support during my studies.

Finally, I would like to thank the Department of Computer Science at Oklahoma State University for supporting during my studies.

TABLE OF CONTENTS

Chapter	Page
Literature Review -----	1
Postemergence Herbicide Application -----	1
Database Models -----	1
Hierarchical Database Model -----	2
Network Database Model -----	3
Relational Database Model -----	5
Visual Basic Communicates With Databases -----	6
Introduction -----	7
Objectives -----	8
Database Design -----	8
1. Data Collection -----	8
2. Database Structure -----	9
3. Database Tables -----	11
4. Relational Schema -----	13
5. Table Structure -----	14
6. Implementation and Loading -----	15
1). Create Tables -----	15
2). Create Queries -----	16
Connecting the Visual Basic GUI to the Database -----	20
1. Add New Records -----	21
2. Delete Records -----	25
3. Find Records -----	27
4. Update Records -----	28
5. Show Tables -----	31
6. Select Herbicides -----	32
7. Exit -----	33
Optimal Herbicide Selection -----	34

Chapter	Page
Summary, Conclusions and Future Work -----	37
Summary -----	37
Conclusions -----	37
Future Work-----	38
Visual Basic 6.0 Form and Code -----	39
1. Form frminterface and Code -----	39
2. Form frmAdd and Code-----	42
3. Form frmAddCrop and Code -----	43
4. Form frmAddWeed and Code -----	43
5. Form frmAddHerb and Code -----	44
6. Form frmAddOthers and Code -----	45
7. Form frmDelete and Code-----	46
8. Form frmFindByKey and Code-----	52
9. Form frmUpdate and Code -----	55
10. Form frmUpdateCrop and Code-----	56
11. Form frmUpdateWeed and Code-----	58
12. Form frmUpdateHerb and Code-----	61
13. Form frmShowtables and Code-----	63
14. Form frmShowCrop and Code-----	65
15. Form frmShowWeed and Code-----	65
16. Form frmShowHerb and Code-----	66
17. Form frmShowRoCrop and Code -----	66
18. Form frmShowCompaProd and Code -----	67
19. Form frmShowUse and Code -----	67
20. Form frmShowControl and Code-----	68
21. Form frmShowAffect and Code -----	68
22. Form frmShowMixWith and Code-----	69
23. Form frmSelectHerb and Code -----	70
Literature Cited-----	93

LIST OF FIGURES

Figure	Page
1. E-R diagram for chemical weed control -----	10
2. Relational Schema -----	13
3. Graphical User Interface for Weed Control -----	20
4. Add Choice form -----	21
5. Add Crop form -----	22
6. Add Weed form -----	22
7. Add Herbicide form -----	23
8. Add Others form -----	24
9. Delete Record form -----	25
10. Find By Key Word form -----	28
11. Update form -----	28
12. Update Crop form -----	29
13. Update Weed form -----	30
14. Update Herbicide form -----	31
15. Show Tables form -----	31
16. Show Crop Table form -----	32
17. Herbicide Selection form -----	32

Literature Review

Postemergence Herbicide Application

At present there are many weed control and management measures such as tillage, crop rotation, chemical control and biological control. Among them, herbicides play an important role in agricultural production. With the determination, establishment and use of different economic weed control thresholds in different crops, postemergence herbicide application will receive increased attention. Postemergence herbicide application refers to those herbicides that are applied after the weed or crop emerges. This practice may provide crop producers an opportunity to treat only those fields with weed densities that are above an economic threshold level, and to control those target weeds with the proper herbicides. Some researchers (Wilkerson, et al, 1991; and Mortense and Coble, 1991) have pointed out that using large quantities of herbicides to control weeds regardless of weed density and economic return is unrealistic and environmentally unacceptable.

In order to have economical and effective control of target weeds and avoid unnecessary herbicide use, postemergence herbicide applications are valuable.

Database Models

A database model refers to a collection of logical constructs used to represent the data structure and the data relationships found within the database. Database models can be divided into two categories: conceptual models and implementation models. Conceptual models focus on the logical nature of the data representation. They are concerned with

what is represented rather than how it is represented. There exist three types of relationships in conceptual database models such as one-to-many relationships, many-to-many relationships and one-to-one relationships. Implementation models place the emphasis on how the data are represented in the database or on how the data structures are implemented. Three types of implementation database models include hierarchical database model, network database model and relational database model (Peter, R and C. Coronel, 1997).

Hierarchical Database Model

A hierarchical database consists of a collection of records that are connected to each other through links. Each record is a collection of fields (attributes), each of which contains only one data value. A link is an association between precisely two records. The hierarchical database model is based on a upside-down tree structure that is composed of a root segment, parent segments, and child segments. The hierarchical database model depicts a set of one-to-many (1:M) relationships between a parent and its children, that is, each parent can have many children and each child has only one parent (Silberschatz, A., H. F. Korth and S. Sudarshan, 1998). The advantages of hierarchical database model are as follows:

- Because all data are held in a common database, data-sharing and security are provided.
- The DBMS creates an environment in which data independence can be maintained, thereby greatly decreasing programming effort and program maintenance.

- Because the child segment is always automatically referenced to its parent, database integrity exists.
- This model can deal with a large volume of data in 1:M relationships.

Unfortunately, the hierarchical model has the following disadvantages:

- This model requires the knowledge of physical level of data storage.
- It is unable to represent relationships that do not conform to the hierarchical 1:M standard.
- Because the hierarchical model does not include ad hoc querying capability, application programming tends to require extensive effort and resources.
- Hierarchical design can also be difficult, therefore the DBMS efficiency is hard to capture.
- Because there are no standard concepts and implementation, the hierarchical model lacks portability

Network Database Model

In many respects the network database model resembles the hierarchical database model; for example, as in the hierarchical model, the user perceives the network database as a collection of records in 1:M relationships. However, quite unlike hierarchical model, a network model's record can have more than one parent ((Ullman, J. D. 1988; Peter, R and C. Coronel, 1997; Silberschatz, A., H. F. Korth and S. Sudarshan, 1998).

In the network model a relationship is called a set. Each set is composed of at least two record types: an owner record that is equivalent to the hierarchical model's parent

and a member record that is equivalent to the hierarchical model's child. The difference between the hierarchical model and the network model is that the a member may have several owners (Peter, R and C. Coronel, 1997).

The network database model has the following advantages:

- The M:N relationships are easier to implement in the network database model than in the hierarchical model.
- The network set defines the relationship between owner and member; its existence allows a program to access an owner record and all the member records within a set, thus yielding greater data access flexibility than was possible with the hierarchical model.
- The network database model enforces database integrity because the user first must define the owner record and then the member.
- The network database model achieves sufficient data independence. Therefore, changes in the data characteristics do not require changes in the applications programs.

The network model's disadvantages include the following:

- Network databases are difficult to design and to use properly. The user must be very familiar with the database structure to take advantage of the system's efficiency.

- Although the network model achieves data independence, it does not produce structural independence. Therefore, it is difficult to make changes in a database.
- The network model has a very complex structure from the application programmer's point of view. Consequently, the programmer must be very familiar with the internal structures in order to access the database.

Relational Database Model

The relational database model is the current database standard. The user/designer need only be concerned with the logical view of the database; the details of physical storage, access paths, and data structures are managed by the (RDBMS). Therefore relational database design becomes much simpler than hierarchical or network models. Relational database is perceived by the user as a collection of tables in which data are stored. Each table consists of series of row/column intersections. Tables (or relations) are related to each other by sharing a common entity characteristic. The relationship type is often shown in a relational schema. A table yields complete data and structural independence because it is a purely logical structure (Ullman, J. D. 1988 and 1989; Peter, R and C. Coronel, 1997; Silberschatz, A., H. F. Korth and S. Sudarshan, 1998).

The relational database model has the following advantages:

- Because the relational model achieves both data independence and structural independence, it becomes much easier to design the database and to manage its contents.

- Less programming effort is required because the relational database has a very powerful query language called Structured Query Language (SQL), which makes ad hoc queries possible.

The disadvantages of relational database model are:

- RDBMS requires substantial hardware and operating system overhead.

Visual Basic Communicates With Databases

Visual Basic communicates with databases through the data control. Data control can read, modify, delete, and add records to databases. The visual boxes such as Text box, check box, image, label, picture box, list box, combo box, data bound list box, data bound combo box and FlexGrid can be bound to a data control (Schneider, D. I. 1998).

Visual Basic is the programming language provided with Access. Visual Basic instructions can respond to events that occur with the objects in a database. We can do anything with Visual Basic that we can do with macros in Access, but Visual Basic gives us more control over commands and objects than we have with macros. With Visual Basic we can create our own functions to perform special calculations.

We can change an object's properties dynamically, for example, Visual Basic code can change the color on a form when the user enters a specific field value (Adamski, J. J., C. Hommel and K. T. Finnegan, 1998; Schneider, D. I. 1998).

Introduction

Weed is one of the important factors that decrease crop yield. It has been proved that it is impossible to wipe out weeds completely (Li, S. and Z. Gao. 1990). Destroying weeds excessively has brought about a series of ecological problems such as soil erosion, herbicide resistance, herbicide residues in ground and surface water, toxicity to non-target organisms and potential for contamination of the environment (Gould, 1995, and Maxwell et al, 1990). These problems have gained worldwide attention, and a number of highly effective, selective postemergence herbicides have been developed for use in different crop fields to control weeds (Wilkerson et al, 1991). These new chemicals provide the opportunity to shift from a preventive to a remedial approach to weed control. Traditional weed control systems have included the use of the preplant incorporated or preemergence herbicides that must be applied before weeds emerge. By delaying a herbicide use decision until after crop and weed emergence, a producer can assess the weed problems more accurately and reduce the risk of applying unnecessary or inappropriate chemicals (Hall et al, 1992, Radosevich, 1987 and Van Acker et al, 1993).

Choosing a suitable herbicide is a complex task in weed management (Stigliani et al, 1996). Prior to the present time the needed data were stored inefficiently, experimental information is often scattered over notebooks, computer files, and computers, while some could be stored only in the memories of people involved. Such information format can be the limiting factor in the weed control because these fragmented and often incompletely documented data are difficult to manipulate in an automated fashion due to the variety of storage formats employed (Van Evert et al, 1991

and Muelier-Warrant, 1991). Furthermore, the structure of the available information does not facilitate the comparison of different herbicides. Therefore, developing a relational database for decision support on chemical weed control is necessary.

Objectives

The Objectives of the project described in this thesis are:

- To design a relational database structure and organize data reasonably and efficiently.
- To provide access to any important information related to relations among weeds, crops and herbicides.
- To connect the Visual Basic GUI to the database and provide a neat, user-friendly method for farmers and agricultural extension workers to access the databases needed for weed control and modify databases according to the latest research results.
- To provide weed scientists and farmers enough information to make decisions on herbicide selection and provide optimal weed control plan relative to various crop and weed growth stages.

Database Design

1. Data Collection The needed data are obtained from relevant books, experimental reports and check with some agricultural extension specialists (Humbyrg, N. E., et al. 1989; Regehr, D. L., et al. 1995; Peterson, D. L. and D. L. Regihr, 1995).

2. Database Structure Database structure should be developed emulating the criteria used by an experienced consultant, furthermore, the herbicide choice can only be the optimum one by considering all the biological (of crop and weeds), biochemical (interaction herbicide/plant) and physical (interaction herbicide/soil or climate) factors involved in a specific situation (Stigliani et al, 1996). Therefore, the relationships among crop, weed, herbicide, crop growth period, susceptible weed growth period and environmental conditions are considered when creating database. In order to assist farmers and agricultural extension workers in making decisions on herbicide selection and optimization for mixed weed infestations at various crop and weed growth stages, data are organized into the E-R diagram shown Figure 1.

The E-R diagram above indicates that the relationships between crop and herbicide, herbicide and weed, herbicide and rotational_crop, herbicide and compatible_product are many-to-many relationships, That is, each crop field can use different herbicides to control weeds, and each herbicide can be used in different crop field; one kind of herbicide can control many kinds of weeds, each weed can be controlled by many kinds of herbicides; each herbicide can affect several kinds of following season's crop growth, and a subsequent crop can be affected by different herbicides; each herbicide can mix with several kinds of other products to control weeds, and each product can mix with other different herbicides.

A relational database can be used to meet these requirements because a relational database can store a large amount of data, deal with minimal values, record data without connections and perform links among data values. Besides these, a relational database

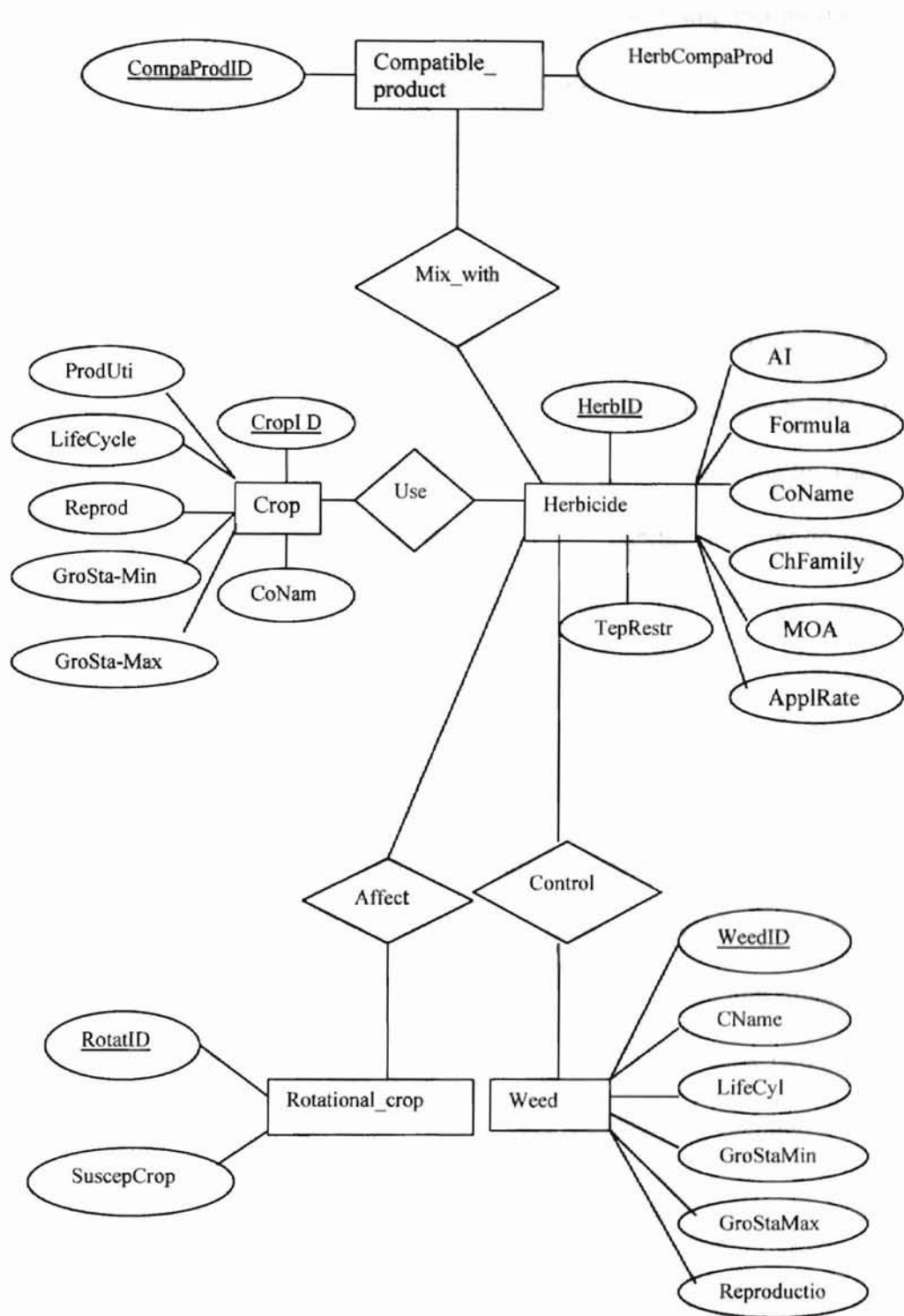


Figure 1. E-R diagram for chemical weed control

model requires data independence and structural independence, Using this model, it is easy to design the database and to manage its contents, and it has powerful and flexible query capability (Ullman, J. D. 1988 and 1989; Peter, R. and C. Coronel, 1997: Ramakrishnan, R, 1998; Silberschatz, A., H. F. Korth and S. Sudarshan, 1998; Zhang, L. and K. Liu, 1993; Codd, E. F. 1970 & 1979; Chen P.P. 1976 and Bernstein P.A. 1976).

Other database models such as the hierarchical database model or the network database model are not suitable for these needs. In a hierarchical database model a tree structure is represented as a hierarchical path on the computer's storage media, each parent can have many children, but each child has only one parent. This one-to-many relationship restriction does not fit our studies (Peter, R. and C. Coronel, 1997 and Silberschatz, A., H. F. Korth and S. Sudarshan, 1998). Although network database model allows relationships among the records are decomposed into a series of sets, easy to implement M:N relationships and have sufficient data independence, this model is difficult to design and use properly, difficult to make changes in a database, it has very complex structure from the application programmer's point of view and complex navigational data access environment (Peter, R. and C. Coronel, 1997 and Silberschatz, A., H. F. Korth and S. Sudarshan, 1998). Therefore, in this studies, these two kinds of models are not used. We use only the relational database model to create and design our weed control database.

3. Database Table In order to avoid data redundancy, data is divided into the following tables:

Crop table

Crop ID-number, Common-name, Life-cycle, Minimal-growth-stage, Maximal-growth-stage, Reproduction, Product utilization.

Weed table

Weed ID-number, Common-name, Life-cycle, Minimal-growth-stage, Maximal-growth-stage, Reproduction.

Herbicide table

Herbicide ID number, Common-name, Chemical-family, Formulation, Mode-of-action, Active-ingredient, Application-rates, Temperature restrictions (Min or Max).

Rotational Crop table

Rotation ID number, Susceptible crop.

Compatible product table

Compatible product ID number, Herbicide compatible product.

Use table

Crop ID number, Herbicide ID number.

Control table

Herbicide ID number, Weed ID number.

Affect table

Herbicide ID number, Rotational crop ID number.

Mix with table

Herbicide ID number, Compatible product ID number.

4. Relational Schema

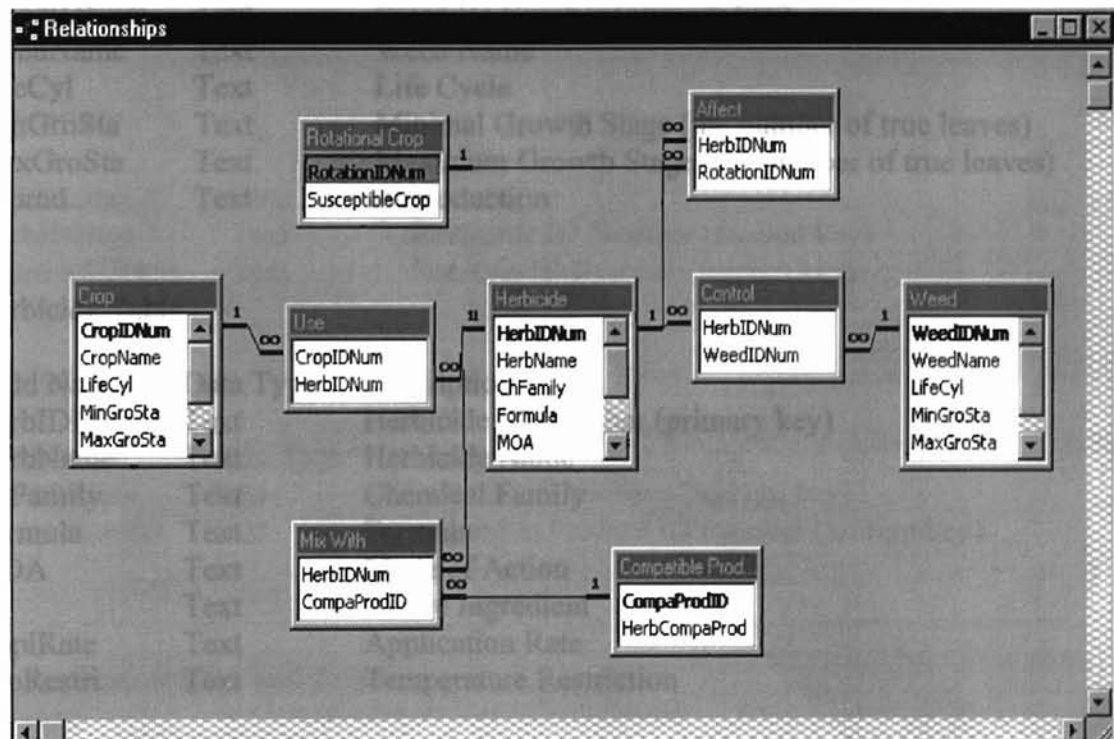


Figure 2. Relational Schema

5. The Table Structure:

Crop Table

Field Name	Data Type	Description
CropIDNum	Text	Crop ID Number (primary key)
CropName	Text	Crop Name
LifeCyl	Text	Life Cycle
MinGroSta	Text	Minimal Growth Stage (the number of true leaves)
MaxGroSta	Text	Maximum Growth Stage (the number of true leaves)
Reprod	Text	Reproduction
ProUtil	Text	Production Utilization

Weed Table

Field Name	Data Type	Description
WeedIDNum	Text	Weed ID Number (primary key)
WeedName	Text	Weed Name
LifeCyl	Text	Life Cycle
MinGroSta	Text	Minimal Growth Stage (the number of true leaves)
MaxGroSta	Text	Maximum Growth Stage (the number of true leaves)
Reprod	Text	Reproduction

Herbicide Table

Field Name	Data Type	Description
HerbIDNum	Text	Herbicide ID Number (primary key)
HerbName	Text	Herbicide Name
ChFamily	Text	Chemical Family
Formula	Text	Formula
MOA	Text	Mode of Action
AI	Text	Active Ingredient
ApplRate	Text	Application Rate
TepRestri	Text	Temperature Restriction

Rotational Crop Table

Field Name	Data Type	Description
RotationIDNum	Text	Rotational Crop ID Number (primary key)

SusceptibleCrop	Text	Susceptible Crop
-----------------	------	------------------

Compatible Product Table

Field Name	Data Type	Description
CompaProdID	Text	Compatible Product ID Number (primary key)
HerbCompaProd	Text	Herbicide Compatible Product

Use Table

Field Name	Data Type	Description
CropIDNum	Text	Crop ID Number (foreign key)
HerbIDNum	Text	Herbicide ID Number (foreign key)

Control Table

Field Name	Data Type	Description
HerbIDNum	Text	Herbicide ID Number (foreign key)
WeedIDNum	Text	Weed ID Number (foreign key)

Affect Table

Field Name	Data Type	Description
HerbIDNum	Text	Herbicide ID Number (foreign key)
RotationIDNum	Text	Rotation ID Number (foreign key)

Mix With Table

Field Name	Data Type	Description
HerbIDNum	Text	Herbicide ID Number (foreign key)
CompaProdID	Text	Compatible Product ID Number (foreign key)

6. Implementation and Loading

1). Create Tables:

- Open Microsoft Access 97
- Open Blank Database

- Create Database Name “Dbgao
- Create the table structure using Design view
- Specify the primary key and foreign key in different tables
- Save the table structure
- Input data into tables
- Create relationships among tables

2). Create Queries:

- cornherb: To find the crop name that is corn, herbicide name. Start a new query, select the Crop table and Herbicide table, add field CropName and HerbName in query, specify criteria CropName=”corn”, then click RUN button.
- cornherbweed: To find the crop name that is corn, herbicide name, and weed name. Start a new query, select the Crop table, Herbicide table and Weed table, add field CropName, HerbName and WeedName in query, specify criteria CropName=”corn”, then click RUN button.
- sorghumherb: To find the crop name that is sorghum, herbicide name. Start a new query, select the Crop table and Herbicide table, add field CropName and HerbName in query, specify criteria CropName=”sorghum”, then click RUN button.
- sorghumherbweed: To find the crop name that is sorghum, herbicide name, and weed name. Start a new query, select the Crop table, Herbicide table and Weed table, add field CropName, HerbName and WeedName in query, specify criteria CropName=”sorghum”, then click RUN button.

- soyherb: To find the crop name that is soybean, herbicide name. Start a new query, select the Crop table and Herbicide table, add field CropName and HerbName in query, specify criteria CropName="soybean", then click RUN button.
- soyherbweed: To find the crop name that is soybean, herbicide name, and weed name. Start a new query, select the Crop table, Herbicide table and Weed table, add field CropName, HerbName and WeedName in query, specify criteria CropName="soybean", then click RUN button.
- sunflowerherb: To find the crop name that is sunflower, herbicide name. Start a new query, select the Crop table and Herbicide table, add field CropName and HerbName in query, specify criteria CropName="sunflower", then click RUN button.
- sunflowerherbweed: To find the crop name that is sunflower, herbicide name, and weed name. Start a new query, select the Crop table, Herbicide table and Weed table, add field CropName, HerbName and WeedName in query, specify criteria CropName="sunflower", then click RUN button.
- oatsherb: To find the crop name that is oats, herbicide name. Start a new query, select the Crop table and Herbicide table, add field CropName and HerbName in query, specify criteria CropName="oats", then click RUN button.
- oatsherbweed: To find the crop name that is oats, herbicide name, and weed name. Start a new query, select the Crop table, Herbicide table and Weed

table, add field CropName, HerbName and WeedName in query, specify criteria CropName="oats", then click RUN button.

- wheatherb: To find the crop name that is wheat, herbicide name. Start a new query, select the Crop table and Herbicide table, add field CropName and HerbName in query, specify criteria CropName="wheat", then click RUN button.
- wheatherbweed: To find the crop name that is wheat, herbicide name, and weed name. Start a new query, select the Crop table, Herbicide table and Weed table, add field CropName, HerbName and WeedName in query, specify criteria CropName="wheat", then click RUN button.
- peanutherb: To find the crop name that is peanut, herbicide name. Start a new query, select the Crop table and Herbicide table, add field CropName and HerbName in query, specify criteria CropName="peanut", then click RUN button.
- peanutherbweed: To find the crop name that is peanut, herbicide name, and weed name. Start a new query, select the Crop table, Herbicide table and Weed table, add field CropName, HerbName and WeedName in query, specify criteria CropName="peanut", then click RUN button.
- alfalfaherb: To find the crop name that is alfalfa, herbicide name. Start a new query, select the Crop table and Herbicide table, add field CropName and HerbName in query, specify criteria CropName="alfalfa", then click RUN button.

- alfalfaherbweed: To find the crop name that is alfalfa, herbicide name, and weed name. Start a new query, select the Crop table, Herbicide table and Weed table, add field CropName, HerbName and WeedName in query, specify criteria CropName="alfalfa", then click RUN button.

Connecting the Visual Basic GUI to the Databases

Visual basic is the programming language provided with Access (Adamski, J. J., C.Hommel and K. T. Finnegan. 1997). It communicates with databases through the data control (Schneider, D. I. 1998 and Schneider, D. I. 1999). In order to provide a neat, user-friendly method for farmer and agricultural extension workers to access the databases needed for weed control (Caldeira, C. P. and P. A. Pinto. 1998 and Pan, X., Hesketh, J. D. and M. G. Huck. 1998) and modify databases according to latest research results, Twenty-three forms have been created. One of them is a main form which is called Graphical User Interface For Weed Control. The main form contains 7 command buttons: **Add New Records**, **Delete Records**, **Update Records**, **Find Records**, **Show Tables**, **Select Herbicide**, and **Exit** (Figure 3).

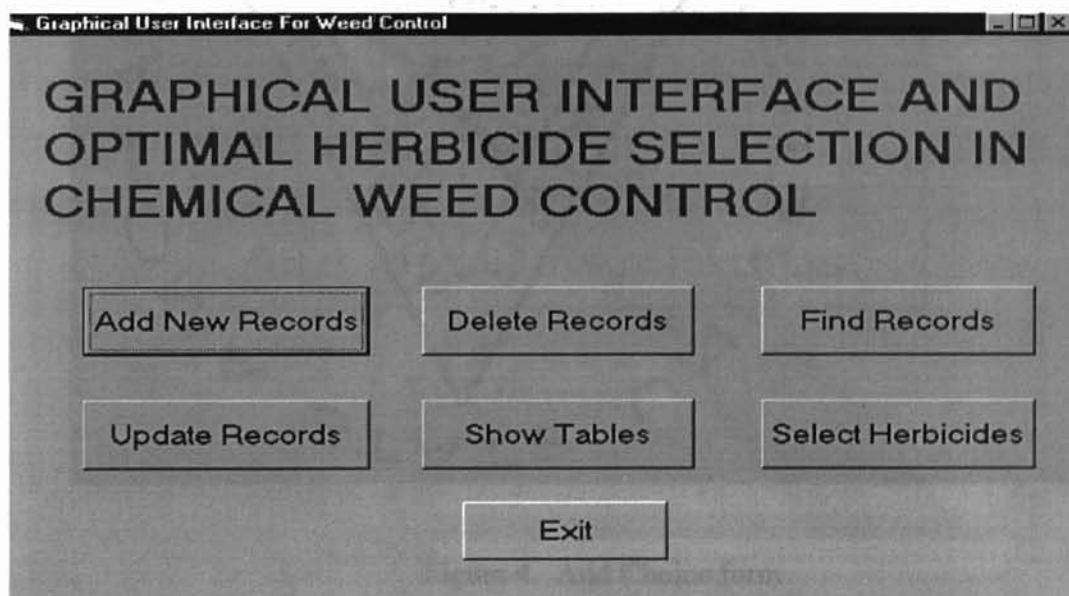


Figure 3. Graphical User Interface for weed control

If the first six buttons are clicked respectively, we are led to other forms to perform add new records, delete records, update records, find records, show the contents of different tables, and select optimal herbicides functions. If we want to stop running the program, we just click **Exit** command button.

1. **Add New Records:** If **Add New Records** button is clicked, we are led to “Add Choice” form which contains five command buttons: **Add Into Crop Table**, **Add Into Weed Table**, **Add Into Herbicide Table**, **Add Into Other Tables** and **Return** (Figure 4). The first four command buttons are linked to other forms to direct us to add information into different tables and **Return** button lets us to return to main menu.

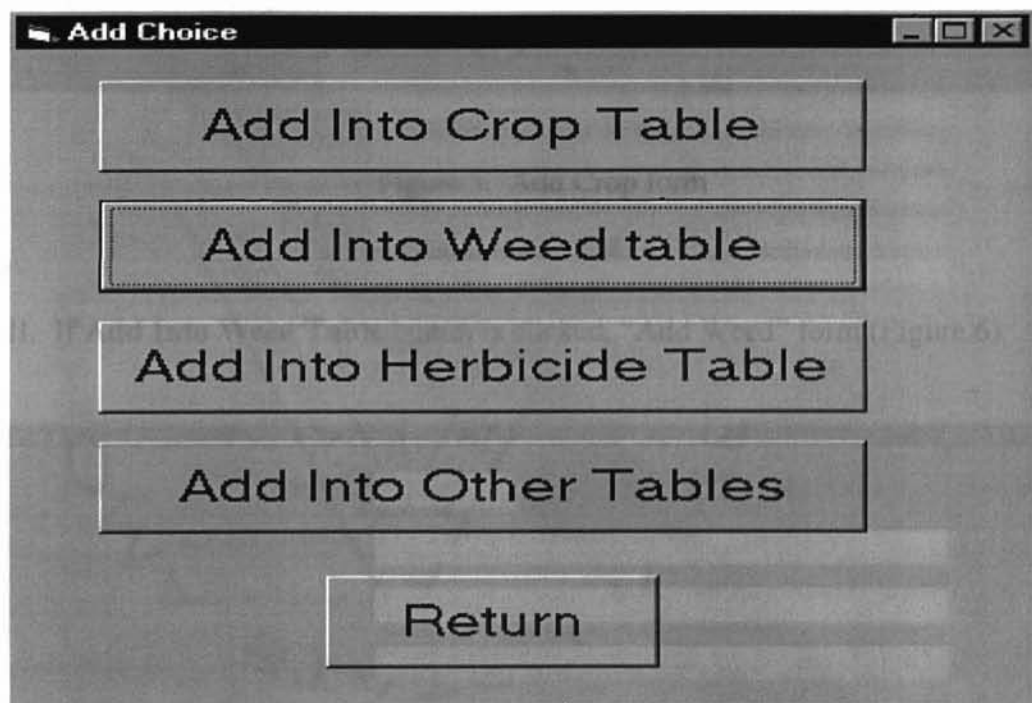
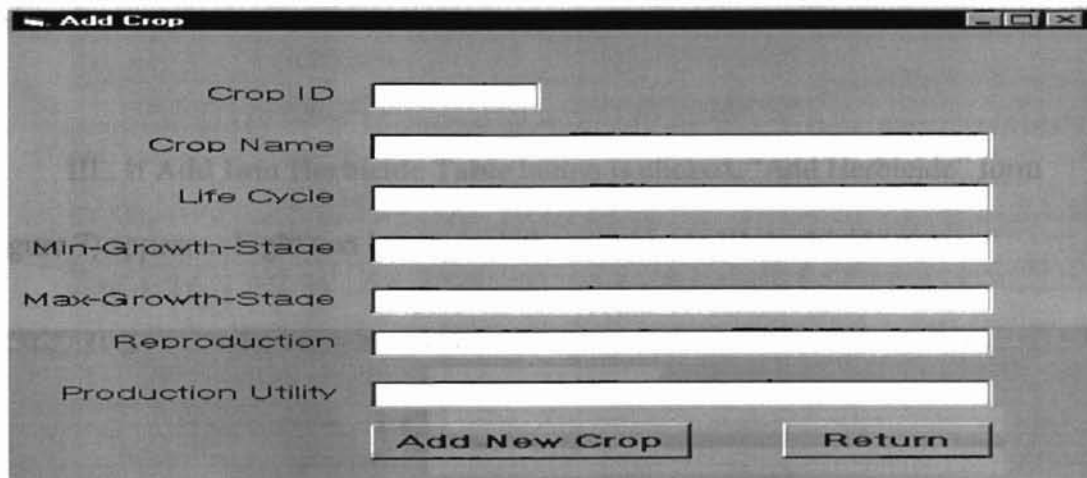


Figure 4. Add Choice form

I. If **Add Into Crop Table** button is clicked, “Add Crop” form (Figure 5) appears. Seven text boxes, including crop ID, crop name, life cycle, minimal growth

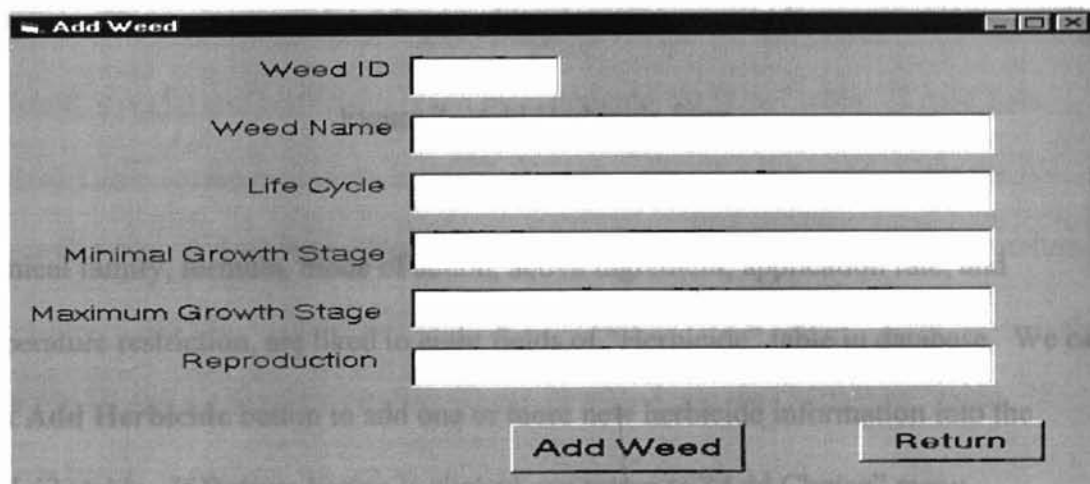
stage, maximum growth stage, reproduction, and production utilization, are linked to seven fields of "Crop" table in database, respectively. We can click **Add New Crop** button to add one or more new crop information into the crop table. If **Return** button is clicked, we return to "Add Choice" menu.



The screenshot shows a window titled "Add Crop". Inside the window, there are seven text input fields arranged vertically, each with a label to its left: "Crop ID", "Crop Name", "Life Cycle", "Min-Growth-Stage", "Max-Growth-Stage", "Reproduction", and "Production Utility". At the bottom of the window, there are two buttons: "Add New Crop" and "Return".

Figure 5. Add Crop form

II. If **Add Into Weed Table** button is clicked, "Add Weed" form (Figure 6)

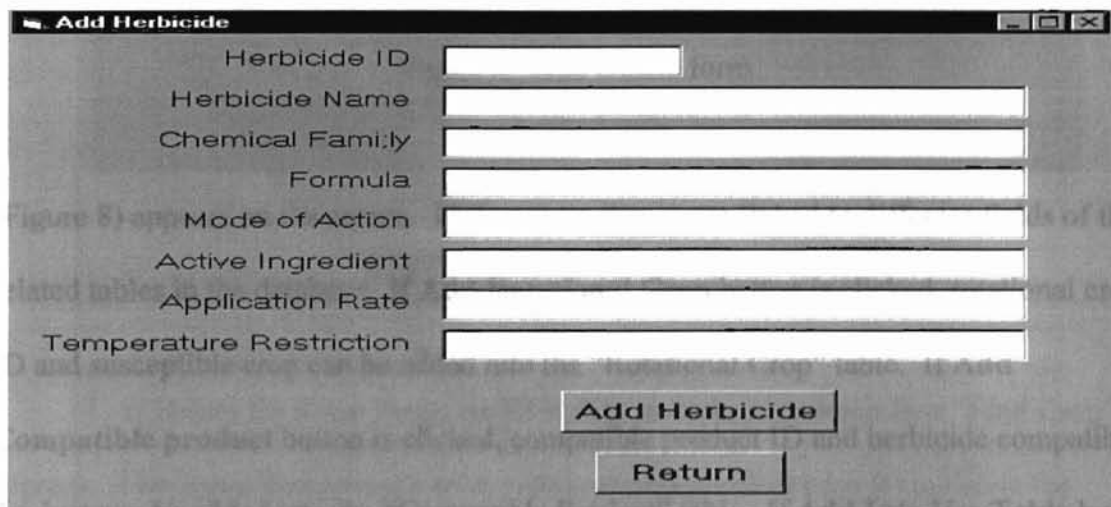


The screenshot shows a window titled "Add Weed". Inside the window, there are six text input fields arranged vertically, each with a label to its left: "Weed ID", "Weed Name", "Life Cycle", "Minimal Growth Stage", "Maximum Growth Stage", and "Reproduction". At the bottom of the window, there are two buttons: "Add Weed" and "Return".

Figure 6. Add Weed form

appears. Six text boxes, including weed ID, weed name, life cycle, minimal growth stage, maximum growth stage, and reproduction, are linked to six fields of “Weed” table in database, respectively. We can click **Add Weed** button to add one or more new weed information into the weed table. If **Return** button is clicked, we return to “Add Choice” menu.

III. If **Add Into Herbicide Table** button is clicked, “Add Herbicide” form (Figure 7) appears. Eight text boxes, including herbicide ID, herbicide name,



The image shows a screenshot of a software window titled "Add Herbicide". Inside the window, there are eight text input fields arranged vertically, each preceded by a label: "Herbicide ID", "Herbicide Name", "Chemical Family", "Formula", "Mode of Action", "Active Ingredient", "Application Rate", and "Temperature Restriction". Below these fields, there are two buttons: "Add Herbicide" and "Return". The window has a standard Windows-style title bar with minimize, maximize, and close buttons.

Figure 7. Add Herbicide form

chemical family, formula, mode of action, active ingredient, application rate, and temperature restriction, are linked to eight fields of “Herbicide” table in database. We can click **Add Herbicide** button to add one or more new herbicide information into the herbicide table. If **Return** button is clicked, we return to “Add Choice” menu.

IV. If **Add Into Other Tables** button is clicked, then the “Add Others” form

Field	Field	Action
Rotation Crop ID	Susceptible Crop	Add Rotational Crop
Compatible product ID	Herb Compatible	Add Compatible Product
Crop ID	Herbicide ID	Add Into Use Table
Herbicide ID	Weed ID	Add Into Control Table
Herbicide ID	Rotation ID	Add Into Affect Table
Herbicide ID	Compatible product ID	Add Into Mix With Table

Return

Figure 8. Add Others form

(Figure 8) appears on the screen. Different textboxes are linked to different fields of the related tables in the database. If **Add Rotational Crop** button is clicked, rotational crop ID and susceptible crop can be added into the “Rotational Crop” table. If **Add Compatible product** button is clicked, compatible product ID and herbicide compatible product can be added into the “Compatible Product” table. If **Add Into Use Table** button is clicked, crop ID and herbicide ID can be added into the “Use” table. If **Add Into Control Table** button is clicked, herbicide ID and weed ID can be added into the “Control” table. If **Add Into Affect Table** button is clicked, herbicide ID and rotation ID can be added into the “Affect” table. If “**Add Into Mix With Table**” button is clicked, herbicide ID and Compatible product ID can be added into the “Mix With” table. If **Return** button is clicked, we return to “Add Choice” menu.

2. **Delete Records:** If **Delete Records** button is clicked, we are led to “Delete Records” form which contains ten command buttons (Figure 9).

The screenshot shows a software window titled "Delete Records". Inside the window, the text "DELETE FROM ENTITY TABLE" is centered at the top. Below this, there are five rectangular buttons arranged in two rows: "Delete By Crop Name Or ID", "Delete By Weed Name Or ID", "Delete By Herbicide Name Or ID", "Delete By Rotational Crop Name Or ID", and "Delete By Compatible Product Name Or ID". Below these, the text "DELETE FROM RELATION TABLE" is centered. Under this, there are four more rectangular buttons: "Delete Records From Use Table", "Delete Records From Control Table", "Delete Records From Affect Table", and "Delete Records From Mix With Table". At the bottom center of the window is a single button labeled "Return".

Figure 9. Delete Records form

I. If **Delete By Crop Name or ID** button is clicked, an Input Box “Find Crop” appears. This Input Box prompts us to type into crop name or crop ID to delete the relevant records from the “Crop” table. If no or wrong information is typed into the textbox of Input Box, an appropriate message is given.

II. If **Delete By Weed Name or ID** button is clicked, an Input Box “Find Weed” appears. This Input Box prompts us to type into weed name or weed ID to delete the relevant records from the “Weed” table. If no or wrong information is typed into the textbox of Input Box, a corresponding message is given.

III. If **Delete By Herbicide Name or ID** button is clicked, an Input Box “Find Herbicide” appears. This Input Box prompts us to type into herbicide name or herbicide

ID to delete the relevant records from the “Herbicide” table. If no or wrong information is typed into the textbox of Input Box, then an appropriate message is given.

IV. If **Delete By Rotational Crop Name or ID** button is clicked, an Input Box “Find Rotational Crop” appears. This Input Box prompts us to type into rotational crop name or ID to delete the relevant records from the “Rotational Crop” table. If no or wrong information is typed into the textbox of Input Box, then an appropriate message is given.

V. If **Delete By Compatible Product Name or ID** button is clicked, an Input Box “Find Compatible Product ” appears. This Input Box prompts us to type into compatible product name or ID to delete the relevant records from the “Compatible Product” table. If no or wrong information is typed into the textbox of Input Box, then an appropriate message is given.

VI. If **Delete Records From Use Table** button is clicked, an Input Box “Find Crop” appears. This Input Box prompts us to type crop ID into the text box. Then another Input Box “Find Herbicide” is shown and it guides us to type herbicide ID into the text box. In this way we can delete the relevant records from the “Use ” table. If no or wrong information is typed into the textbox of Input Box, then an appropriate message is given.

VII. If **Delete Records From Control Table** button is clicked, an Input Box “Find Herbicide” appears. This Input Box prompts us to type Herbicide ID into the text box. Then another Input Box “Find Weed ID” is shown and it guides us to type Weed ID into the text box. In this way we can delete the relevant records from the “Control” table.

If no or wrong information is typed into the textbox of Input Box, then an appropriate message is given.

VIII. If **Delete Records From Affect Table** button is clicked, an Input Box “Find Herbicide” appears. This Input Box prompts us to type herbicide ID into the text box. Then another Input Box “Find Rotational Crop ID” is shown and it guides us to type rotational crop ID into the text box to delete the relevant records from the “Affect” table. If no or wrong information is typed into the textbox of Input Box, then an appropriate message is given.

IX. If **Delete Records From Mix With Table** button is clicked, an Input Box “Find Herbicide” appears. This Input Box prompts us to type herbicide ID into the text box. Then another Input Box “Find Compatible Product” is shown and it guides us to type compatible product ID into the text box to delete the relevant records from the “Mix With” table. If no or wrong information is typed into the textbox of Input Box, then an appropriate message is given.

X. If **Return** button is clicked, we return to the menu “Delete Records”.

3. Find Records: If **Find Records** button is clicked, “Find By Key Word” form shows up (Figure 10).

If crop ID or common name is typed into text box and then **Find Crop** button is clicked, the relevant crop information is shown on the picture box. Similarly, if weed ID or common name (or herbicide ID or common name) is typed into the text box and then **Find Weed** (or **Find Herbicide**) button is clicked, the information of weed (or herbicide) is shown on the picture box. If no or wrong information is input into the text box, prompt

Figure 10. Find By Key Word form

or message is given. If **Return** button is clicked, we return to main menu.

4. Update Records: If **Update Records** button is clicked, we are led to “Update” form which contains four command buttons: **Update Crop**, **Update Weed**, **Update Herbicide** and **Return** (Figure 11). The first three command buttons are linked

Figure 11. Update form

to other forms to direct us to perform update and **Return** button lead us to return to main menu.

I. If **Update Crop** button is clicked, “Update Crop” form (Figure 12) appears.

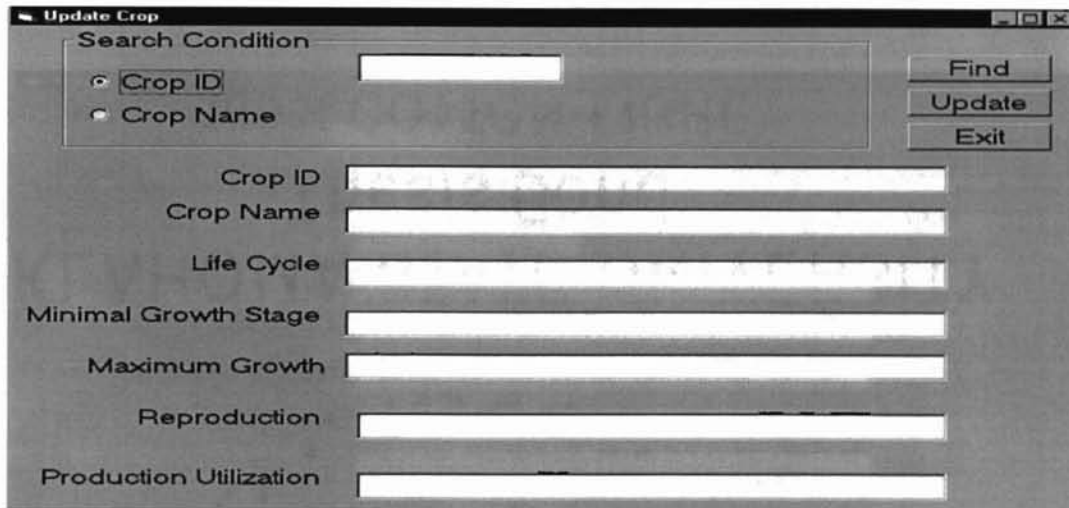


Figure 12. Update Crop form

If we type the search condition (Crop ID or Crop Name) into the text box and click **Find** button, the corresponding crop information record that conform to given condition, including crop ID, crop name, life cycle, minimal growth stage, maximum growth stage, reproduction, and production utilization, are shown on the seven text boxes. After you revise the crop record and then click **Update** button, the current record can be updated successfully. If Exit button is clicked, we return to “Update” form.

II. If **Update Weed** is clicked, “Update Weed” form (figure 13) is shown. If we type search condition (Weed ID or Weed Name) into the text box and click **Find** button, the corresponding weed information record that conform to given condition, including

weed ID, weed name, life cycle, minimal growth stage, maximum growth stage, and reproduction, are shown on the six text boxes. After you revise the weed record and then click **Update** button, the current record can be updated successfully. If Exit button is clicked, we return to “Update” form.

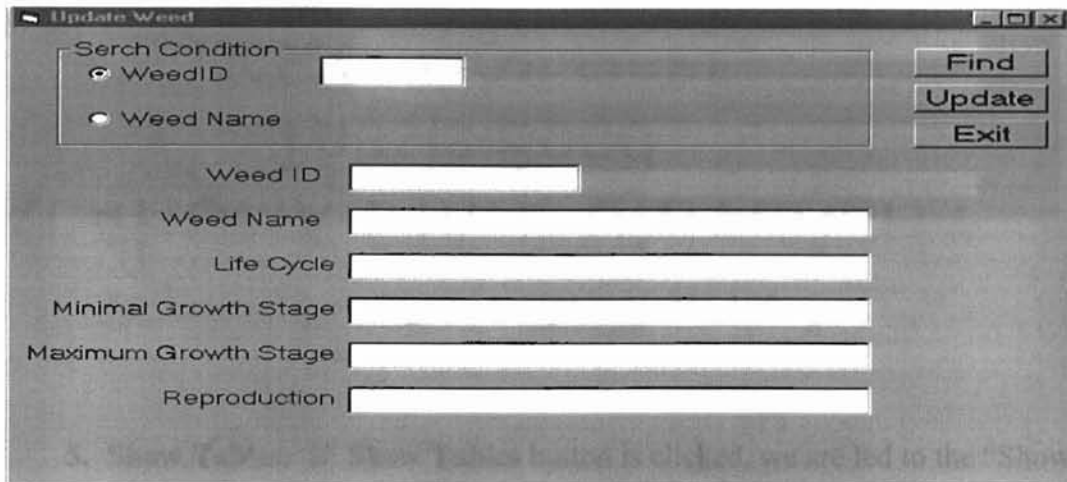


Figure 13. Update Weed form

III. If **Update Herbicide** is clicked, “Update Herbicide” form (figure 14) appears. If we type search condition (Herbicide ID or Herbicide Name) into the text box and click **Find** button, the corresponding herbicide information record that conform to given condition, including herbicide ID, herbicide name, chemical family, formula, mode of action, active ingredient, application rate, and temperature restriction, are shown on the eight text boxes. After you revise the herbicide record and then click **Update** button, the current record can be updated successfully. If Exit button is clicked, we return to “Update” form.

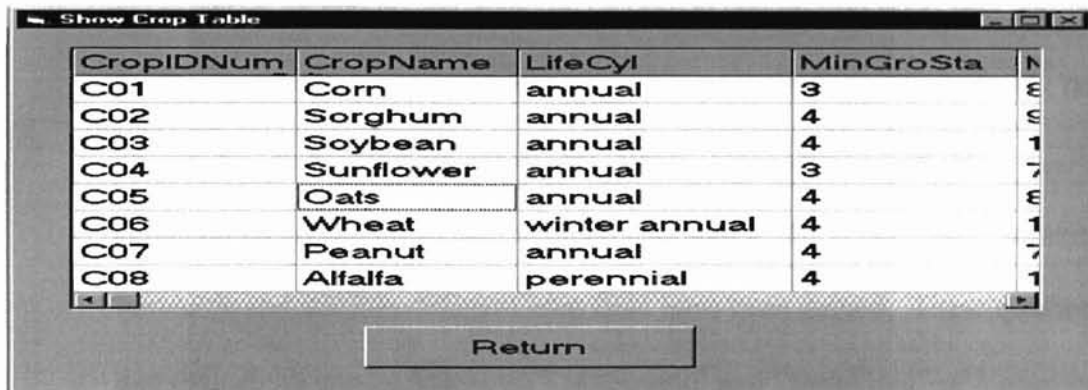
Figure 14. Update Herbicide form

5. Show Tables: If **Show Tables** button is clicked, we are led to the “Show tables” form which contains ten command buttons: **Show Crop Table**, **Show Weed Table**, **Show Herbicide Table**, **Show Rotational Crop Table**, **Show Compatible Product Table**, **Show Use Table**, **Show Control Table**, **Show Affect table**, **Show Mix With Table** and **Return** (Figure 15). The first nine command buttons are linked to other

Figure 15. Show Tables form

forms to show different tables in the database and **Return** button lead us to return to main menu.

If **Show Crop Table** button is clicked, “Show Crop table” form (Figure 16) is shown. We can view the “Crop” table of the database management system. Similarly, we can view other table contents of database management system.

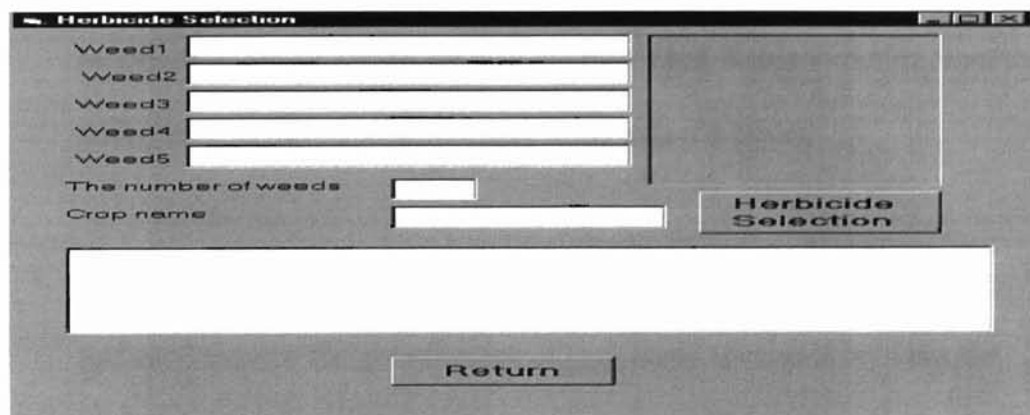


CropIDNum	CropName	LifeCyl	MinGroSta	
C01	Corn	annual	3	8
C02	Sorghum	annual	4	9
C03	Soybean	annual	4	1
C04	Sunflower	annual	3	7
C05	Oats	annual	4	8
C06	Wheat	winter annual	4	1
C07	Peanut	annual	4	7
C08	Alfalfa	perennial	4	1

Return

Figure 16. Show Crop Table form

6. Select Herbicides: If **Select Herbicides** button is clicked, we are led to “Herbicide Selection” form (Figure 17). We can type the designated weeds, the number



Herbicide Selection

Weed1

Weed2

Weed3

Weed4

Weed5

The number of weeds

Crop name

Herbicide Selection

Return

Figure 17. Herbicide Selection form

of weeds, and crop name into the different text boxes and then click **Herbicide Selection** button, the optimal herbicide combination can be found and shown in the list box. If no or wrong information is input into text boxes, proper message is given.

7. **Exit:** When **Exit** button is clicked, the running program terminates.

Optimal Herbicide Selection

Selecting optimal herbicides is our main purpose. Realization of this goal needs the following processes:

1. Through database-driven queries, a list of herbicides that meet the biological and environmental requirements in different crop field can be obtained. The information on which herbicide can be used in which crop field, and which herbicide can control what weeds in specific crop field can be obtained. In the section of database design and establishment, the queries about the relationship between crop and herbicide, as well as the relationship between herbicide and weed in different crop field have been gained. These queries and their fields can be linked to “DataSource” and “DataField” of Visual basic 6.0 through the data control to complete the task of the optimal herbicide selection.
2. Check and ensure that crop growth stages are between the minimum and the maximum growth stage related to the crop herbicide tolerance. This information can be gained through clicking **Find Records** button from the main menu to trace the attributes of different crop tables.
3. The susceptibility to the herbicide is checked for each selected weed species, and ensuring that the growth stage of each weed species is between the minimum and maximum weed growth stages at which the herbicide is

effective. We can get this information by clicking **Find Weed** button on the form of "Find By Key Word".

4. Since the weed flora is heterogeneous, single herbicide is difficult to meet the practical weed control requirements. In general, there are 1 to 4 different weeds that dominate the crop field. In order to meet the requirement that the farmers want to control as many weeds as possible, five different kinds of weeds can be designated in this program. The program can make comparisons among the active ingredients to select herbicides that control all the selected weeds.
5. In order to run a program to find the lowest number of herbicides controlling all the designated weeds quickly, an optimization program is run. This program utilizes a dynamic two by two list of herbicides and weed species. In fact, one herbicide can control several weeds and one weed can be killed by different herbicides. The first algorithm step is to create weed array and herbicide array. Weed array contains the weeds that will be controlled and herbicide array contains the herbicides that can be used in given crop field. The second algorithm step is to set up the matrix that describes the relationship between different herbicides and weeds. If this kind of herbicide can control this kind of weed, we record a "1" on the cross position of the herbicide and the weed. Otherwise, record a "0". The last algorithm step is to check the established matrix and make comparisons to find the lowest

number of herbicides controlling all the designated weeds. During the process of checking the matrix and making the comparisons, the program selects the herbicide that can kill the most weeds alone in the first turn. In the second turn, the program selects another herbicide that can control the most remaining weeds. In this way, the program selects an appropriate herbicide combination for weed control.

Summary, Conclusions and Future Work

Summary

This study uses the relational database model to create and design the weed control database. In order to avoid data redundancy, data is divided into nine tables: Crop table, Weed table, Herbicide Table, Rotational Crop table, Compatible Product table, Use table, Control table, Affect table and Mix With table. The established database provides access to any important information related to relations among weeds, crops and herbicides, and it also provides a good foundation for choosing good herbicide combination in chemical weed control.

This study connects the Visual Basic GUI to the database and provides a neat, user-friendly interface for farmers and agricultural extension workers to access the databases needed for weed control and modify databases according to the latest research results. By clicking relevant keys on the GUI, users can add new records into the database table, delete records from the database, find records from the database, update records in the database, show table's contents of the database, and obtain the lowest number of herbicides controlling all the chosen weeds.

Conclusions

- This study provides access to any important information related to relations among weeds, crops and herbicides.
- This study provides a neat, user-friendly interface for user to use and operate

the database easily, that is, user can freely add new records into the database table, delete records from the database, find records from the database, update records in the database, show table's contents of the database

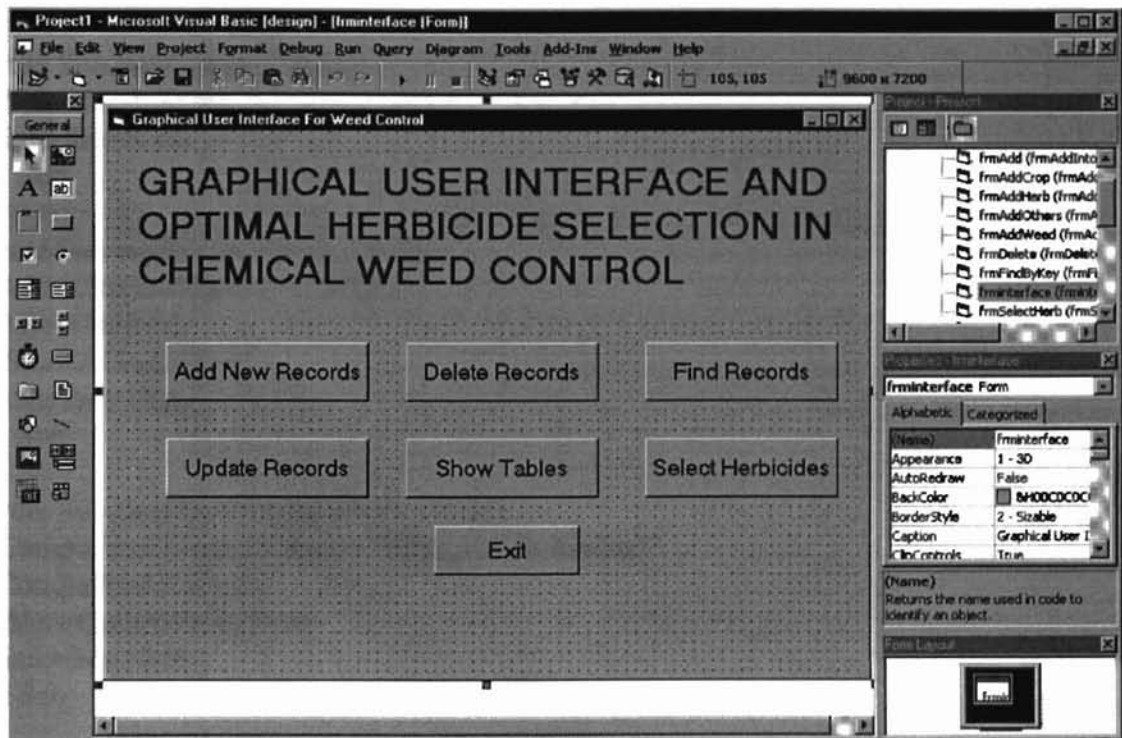
- This study provides an optimization program. Through running this program, the lowest number of herbicides controlling all the chosen weeds is obtained.

Future Work

The results of this study need to be introduced to the farmers and agricultural extension workers as soon as possible with the help of the training classes. The farmers and agricultural extension workers can use this database and user-friendly interface easily and efficiently to use and operate weed control database and get the optimal herbicide combination for weed control.

Visual basic 6.0 Form and Code

1. Form frminterface and Code:



```
Private Sub cmdAdd_Click()  
frmAdd.Show  
frminterface.Hide  
End Sub
```

```
Private Sub cmdDelete_Click()  
frmDelete.Show  
frminterface.Hide  
End Sub
```

```
Private Sub cmdFind_Click()  
frmFindByKey.Show  
frminterface.Hide  
End Sub  
Private Sub cmdQuit_Click()  
End
```

End Sub

```
Private Sub cmdRoCrop_Click()  
datRoCrop.RecordSource = "Rotational Crop"  
datRoCrop.Refresh  
frmShowRoCrop.Show  
frminterface.Hide  
End Sub
```

```
Private Sub cmdSelect_Click()  
frmSelectHerb.Show  
frminterface.Hide  
End Sub
```

```
Private Sub cmdShowAffect_Click()  
datAffect.RecordSource = "Affect"  
datAffect.Refresh  
frmShowAffect.Show  
frminterface.Hide  
End Sub
```

```
Private Sub cmdShowCompaProd_Click()  
datCompaProd.RecordSource = "Compatible Product"  
datCompaProd.Refresh  
frmShowCompaProd.Show  
frminterface.Hide  
End Sub
```

```
Private Sub cmdShowControl_Click()  
datControl.RecordSource = "Control"  
datControl.Refresh  
frmShowControl.Show  
frminterface.Hide  
End Sub
```

```
Private Sub cmdShowCrop_Click()  
datCrop.RecordSource = "Crop"  
datCrop.Refresh  
frmShowCrop.Show  
frminterface.Hide  
End Sub
```

```
Private Sub cmdShowHerbicide_Click()  
datHerb.RecordSource = "Herbicide"  
datHerb.Refresh  
frmShowHerb.Show
```

```
frminterface.Hide  
End Sub
```

```
Private Sub cmdShowMixWith_Click()  
datMixWith.RecordSource = "Mix With"  
datMixWith.Refresh  
frmShowMixWith.Show  
frminterface.Hide  
End Sub
```

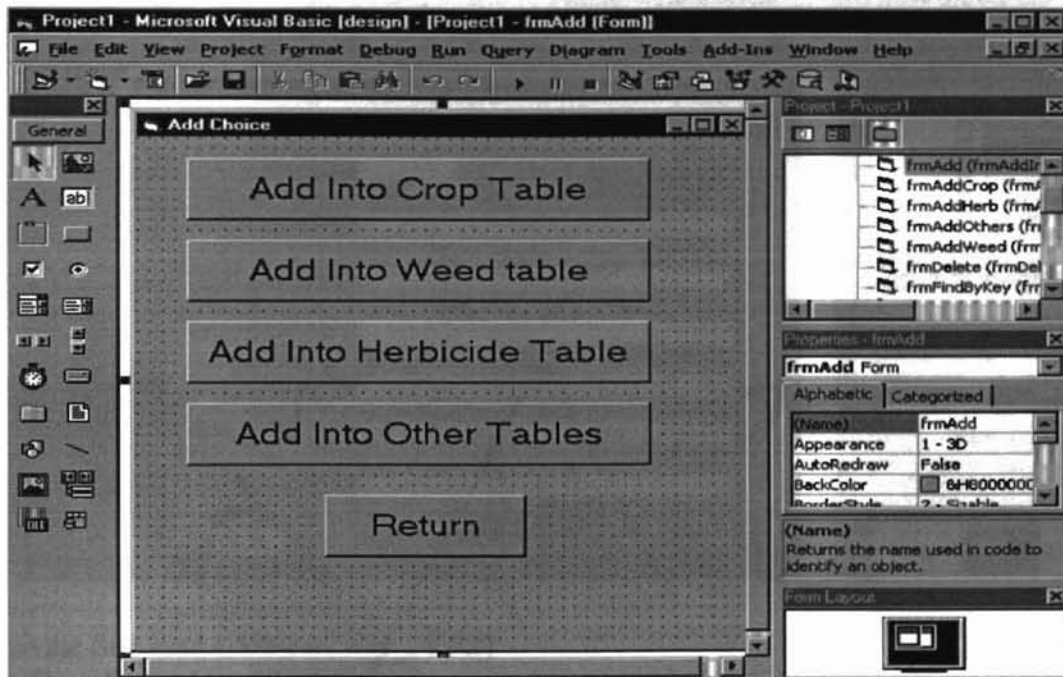
```
Private Sub cmdShowTables_Click()  
frmShowtable.Show  
frminterface.Hide  
End Sub
```

```
Private Sub cmdShowUse_Click()  
datUse.RecordSource = "Use"  
datUse.Refresh  
frmShowUse.Show  
frminterface.Hide  
End Sub
```

```
Private Sub cmdShowWeed_Click()  
datWeed.RecordSource = "Weed"  
datWeed.Refresh  
frmShowWeed.Show  
frminterface.Hide  
End Sub
```

```
Private Sub cmdUpdate_Click()  
frminterface.Hide  
frmUpdate.Show  
End Sub
```

2. Form frmAdd and Code:



```
Private Sub cmdAddIntoCrop_Click()  
frmAdd.Hide  
frmAddCrop.Show  
End Sub
```

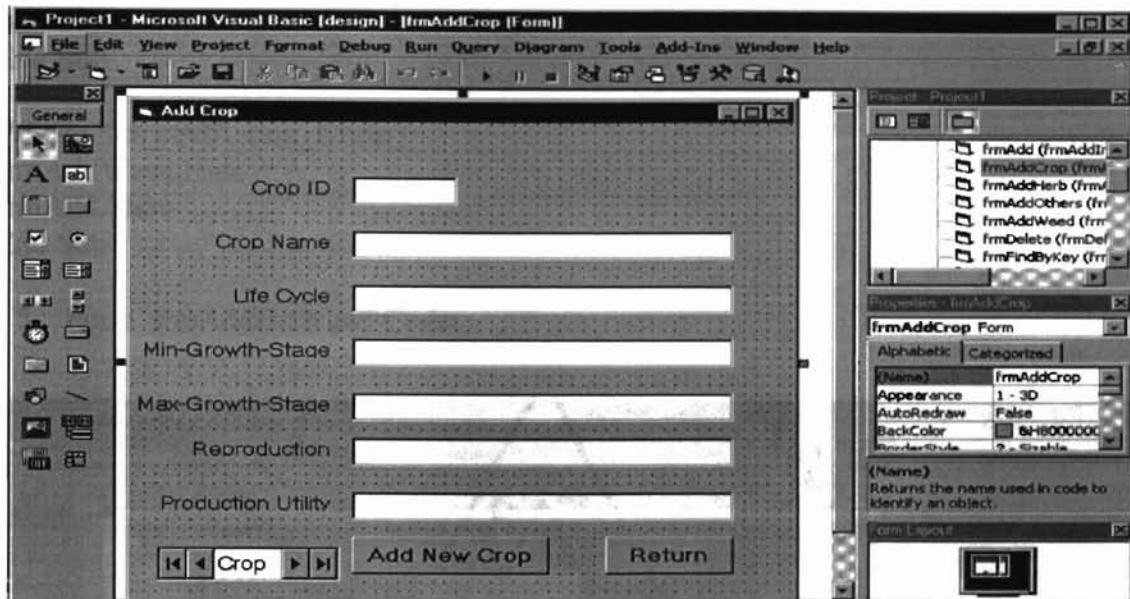
```
Private Sub cmdAddIntoherb_Click()  
frmAdd.Hide  
frmAddHerb.Show  
End Sub
```

```
Private Sub cmdAddIntoOthers_Click()  
frmAddOthers.Show  
frmAdd.Hide  
End Sub
```

```
Private Sub cmdAddIntoWeed_Click()  
frmAdd.Hide  
frmAddWeed.Show vbModal  
End Sub
```

```
Private Sub cmdReturn_Click()  
frmAdd.Hide  
frminterface.Show  
End Sub
```

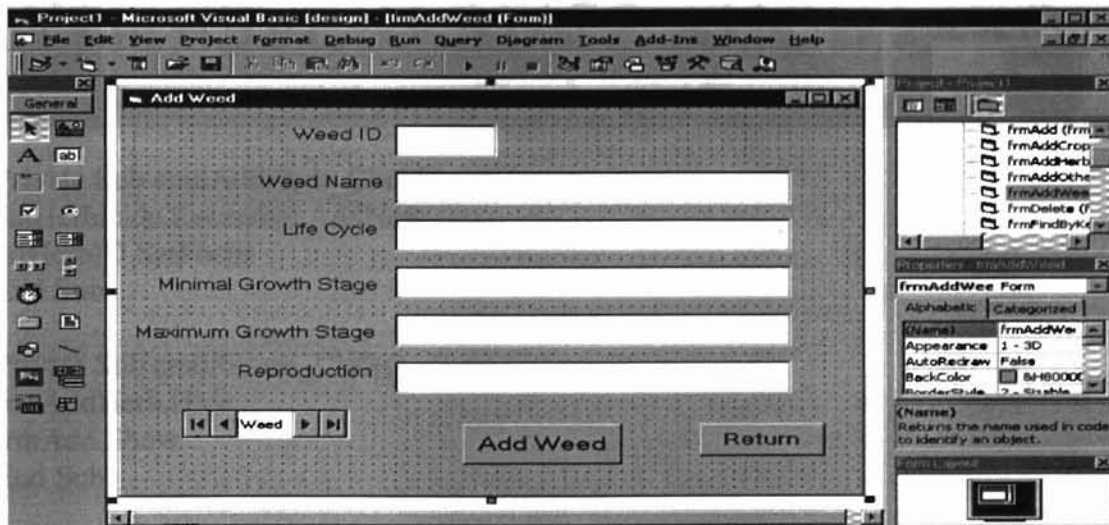
3. Form frmAddCrop and Code:



```
Private Sub cmdAddNewCrop_Click()
    datCrop.Recordset.AddNew
    txtCropIDNum.SetFocus
End Sub
```

```
Private Sub cmdReturn_Click()
    frmAddCrop.Hide
    frmAdd.Show
End Sub
```

4. Form frmAddWeed and Code:



```

Private Sub cmdAddWeed_Click()
datWeed.Recordset.AddNew
txtWeedID.SetFocus
End Sub

```

```

Private Sub cmdReturn_Click()
frmAddWeed.Hide
frmAdd.Show
End Sub

```

5. Form frmAddHerb and Code:



```

Private Sub cmdAddHerb_Click()
datHerbicide.Recordset.AddNew
txtHerbID.SetFocus
End Sub

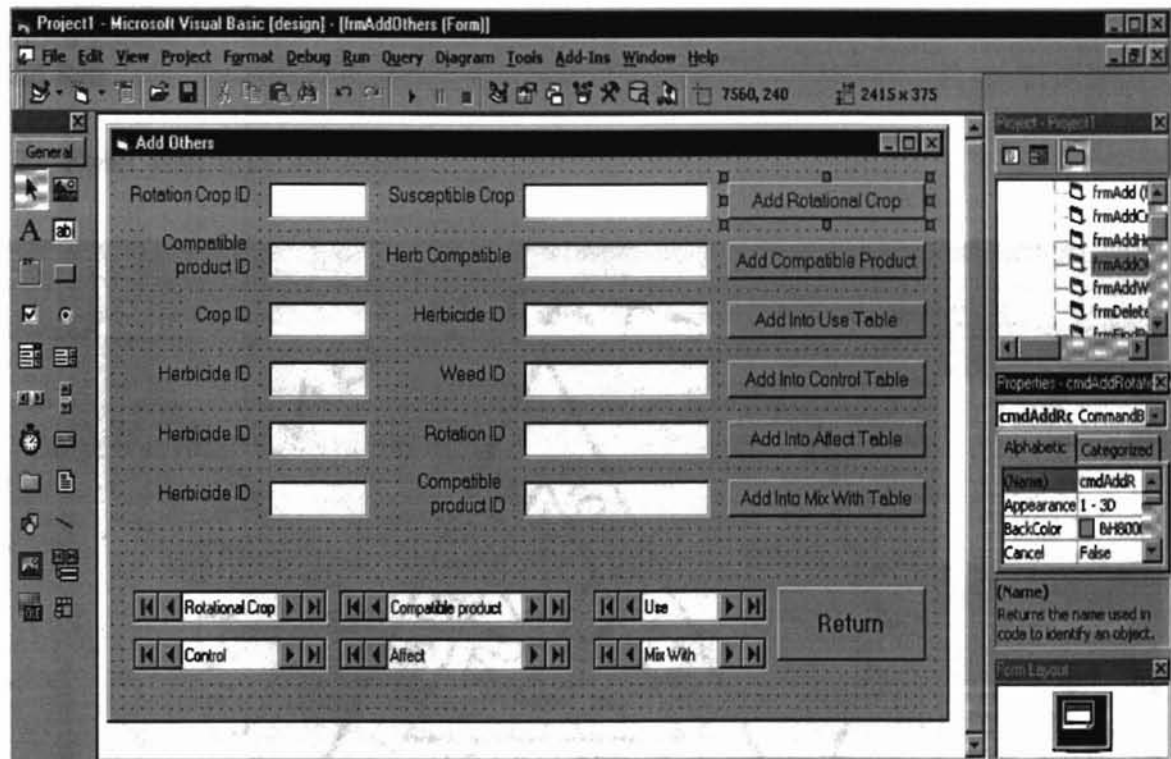
```

```

Private Sub cmdReturn_Click()
frmAddHerb.Hide
frmAdd.Show
End Sub

```

6. Form frmAddOthers and Code:



```
Private Sub cmdAddAffect_Click()
datAffect.Recordset.AddNew
End Sub
```

```
Private Sub cmdAddCompaProd_Click()
datCompatibleProduct.Recordset.AddNew
End Sub
```

```
Private Sub cmdAddIntoControl_Click()
datControl.Recordset.AddNew
End Sub
```

```
Private Sub cmdAddIntoUse_Click()
datUse.Recordset.AddNew
End Sub
```

```
Private Sub cmdAddMixWith_Click()
datMixWith.Recordset.AddNew
End Sub
```

```
Private Sub cmdAddRotationalCrop_Click()
datRotationalCrop.Recordset.AddNew
End Sub
```

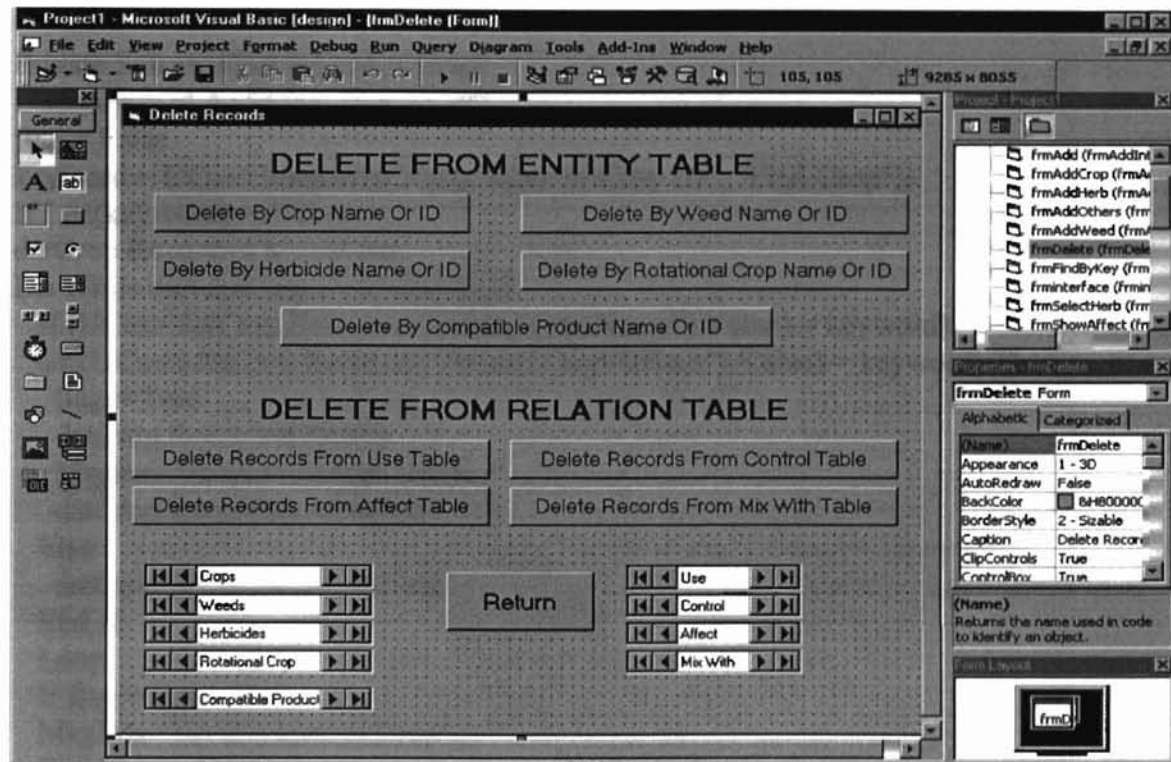


```

Private Sub cmdReturn_Click()
frmAddOthers.Hide
frmAdd.Show
End Sub

```

7. Form frmDelete and Code:



```

Private Sub cmdDeleteCompaProd_Click()
Dim keyword As String, flag As Boolean
flag = False
keyword = UCase(InputBox("Compatible Product Name or ID", "Find Compatible product"))
If Len(keyword) > 0 Then
datCompaProd.Recordset.MoveFirst
Do While Not datCompaProd.Recordset.EOF
If (UCase(datCompaProd.Recordset.Fields("HerbCompaProd").Value) = keyword) _
Or (UCase(datCompaProd.Recordset.Fields("CompaProdID").Value) = keyword) Then
flag = True
datCompaProd.Recordset.Delete
MsgBox "record has been deleted", , ""
datCompaProd.Recordset.MoveNext
Else
datCompaProd.Recordset.MoveNext

```



```

End If
Loop
If flag = False Then
MsgBox "No this kind of compatible product", , ""
End If
Else
MsgBox "Must enter compatible product name or ID", , ""
End If
End Sub

```

```

Private Sub cmdDeleteCrop_Click()
Dim keyword As String, flag As Boolean
flag = False
keyword = UCase(InputBox("Crop Name or Crop ID", "Find Crop"))
If Len(keyword) > 0 Then
datCrop.Recordset.MoveFirst
Do While Not datCrop.Recordset.EOF
If (UCase(datCrop.Recordset.Fields("CropName").Value) = keyword) _
Or (UCase(datCrop.Recordset.Fields("CropIDNum").Value) = keyword) Then
flag = True
datCrop.Recordset.Delete
MsgBox "record has been deleted", , ""
datCrop.Recordset.MoveNext
Else
datCrop.Recordset.MoveNext
End If
Loop
If flag = False Then
MsgBox "No this kind of Crop", , ""
End If
Else
MsgBox "Must enter crop name or crop ID", , ""
End If
End Sub

```

```

Private Sub cmdDeleteFromAffect_Click()
Dim keyword1 As String, keyword2 As String, flag As Boolean
flag = False
keyword1 = UCase(InputBox("Herbicide ID", "Find Herbicide"))
keyword2 = UCase(InputBox("Rotational Crop ID", "Find Rotational Crop"))
If Len(keyword1) > 0 And Len(keyword2) > 0 Then
datAffect.Recordset.MoveFirst
Do While Not datAffect.Recordset.EOF
If (UCase(datAffect.Recordset.Fields("HerbIDNum").Value) = keyword1) _
And (UCase(datAffect.Recordset.Fields("RotationIDNum").Value) = keyword2) Then
flag = True

```

```

datAffect.Recordset.Delete
MsgBox "record has been deleted", , ""
datAffect.Recordset.MoveNext
Else
    datAffect.Recordset.MoveNext
End If
Loop
If flag = False Then
    MsgBox "No this kind of records", , ""
End If
Else
    MsgBox "Must enter Herbicide ID and Rotational Crop ID", , ""
End If
End Sub

```

```

Private Sub cmdDeleteFromControl_Click()
Dim keyword1 As String, keyword2 As String, flag As Boolean
flag = False
keyword1 = UCase(InputBox("Herbicide ID", "Find Herbicide"))
keyword2 = UCase(InputBox("Weed ID", "Find Weed"))
If Len(keyword1) > 0 And Len(keyword2) > 0 Then
datControl.Recordset.MoveFirst
Do While Not datControl.Recordset.EOF
    If (UCase(datControl.Recordset.Fields("HerbIDNum").Value) = keyword1) _
        And (UCase(datControl.Recordset.Fields("WeedIDNum").Value) = keyword2) Then
        flag = True
        datControl.Recordset.Delete
        MsgBox "record has been deleted", , ""
        datControl.Recordset.MoveNext
    Else
        datControl.Recordset.MoveNext
    End If
Loop
If flag = False Then
    MsgBox "No this kind of records", , ""
End If
Else
    MsgBox "Must enter Herbicide ID and Weed ID", , ""
End If
End Sub

```

```

Private Sub cmdDeleteFromMixWith_Click()
Dim keyword1 As String, keyword2 As String, flag As Boolean
flag = False
keyword1 = UCase(InputBox("Herbicide ID", "Find Herbicide"))
keyword2 = UCase(InputBox("Compatible Product ID", "Find Compatible Product"))

```

```

If Len(keyword1) > 0 And Len(keyword2) > 0 Then
datMixWith.Recordset.MoveFirst
Do While Not datMixWith.Recordset.EOF
    If (UCase(datMixWith.Recordset.Fields("HerbIDNum").Value) = keyword1) _
    And (UCase(datMixWith.Recordset.Fields("CompaProdID").Value) = keyword2) Then
        flag = True
        datMixWith.Recordset.Delete
        MsgBox "record has been deleted", , ""
        datMixWith.Recordset.MoveNext
    Else
        datMixWith.Recordset.MoveNext
    End If
Loop
If flag = False Then
    MsgBox "No this kind of records", , ""
End If
Else
    MsgBox "Must enter Herbicide ID and Compatible Product ID", , ""
End If
End Sub

```

```

Private Sub cmdDeleteFromUse_Click()
Dim keyword1 As String, keyword2 As String, flag As Boolean
flag = False
keyword1 = UCase(InputBox("Crop ID", "Find Crop"))
keyword2 = UCase(InputBox("Herbicide ID", "Find herbicide"))
If Len(keyword1) > 0 And Len(keyword2) > 0 Then
datUse.Recordset.MoveFirst
Do While Not datUse.Recordset.EOF
    If (UCase(datUse.Recordset.Fields("CropIDNum").Value) = keyword1) _
    And (UCase(datUse.Recordset.Fields("HerbIDNum").Value) = keyword2) Then
        flag = True
        datUse.Recordset.Delete
        MsgBox "record has been deleted", , ""
        datUse.Recordset.MoveNext
    Else
        datUse.Recordset.MoveNext
    End If
Loop
If flag = False Then
    MsgBox "No this kind of records", , ""
End If
Else
    MsgBox "Must enter crop ID and crop ID", , ""
End If
End Sub

```

```

Private Sub cmdDeleteHerb_Click()
Dim keyword As String, flag As Boolean
flag = False
keyword = UCase(InputBox("Herbicide Name or ID", "Find Herbicide"))
If Len(keyword) > 0 Then
datHerb.Recordset.MoveFirst
Do While Not datHerb.Recordset.EOF
    If (UCase(datHerb.Recordset.Fields("HerbName").Value) = keyword) _
    Or (UCase(datHerb.Recordset.Fields("HerbIDNum").Value) = keyword) Then
        flag = True
        datHerb.Recordset.Delete
        MsgBox "record has been deleted", , ""
        datHerb.Recordset.MoveNext
    Else
        datHerb.Recordset.MoveNext
    End If
Loop
If flag = False Then
    MsgBox "No this kind of herbicide", , ""
End If
Else
    MsgBox "Must enter Herbicide name or Herbicide ID", , ""
End If

End Sub

```

```

Private Sub cmdDeleteRoTationalCrop_Click()
Dim keyword As String, flag As Boolean
flag = False
keyword = UCase(InputBox("Rotational Crop Name or ID ", "Find Rotational Crop"))
If Len(keyword) > 0 Then
datRoCrop.Recordset.MoveFirst
Do While Not datRoCrop.Recordset.EOF
    If (UCase(datRoCrop.Recordset.Fields("SusceptibleCrop").Value) = keyword) _
    Or (UCase(datRoCrop.Recordset.Fields("RotationIDNum").Value) = keyword) Then
        flag = True
        datRoCrop.Recordset.Delete
        MsgBox "record has been deleted", , ""
        datRoCrop.Recordset.MoveNext
    Else
        datRoCrop.Recordset.MoveNext
    End If
Loop
If flag = False Then
    MsgBox "No this kind of Rotational Crop", , ""
End If

```

```

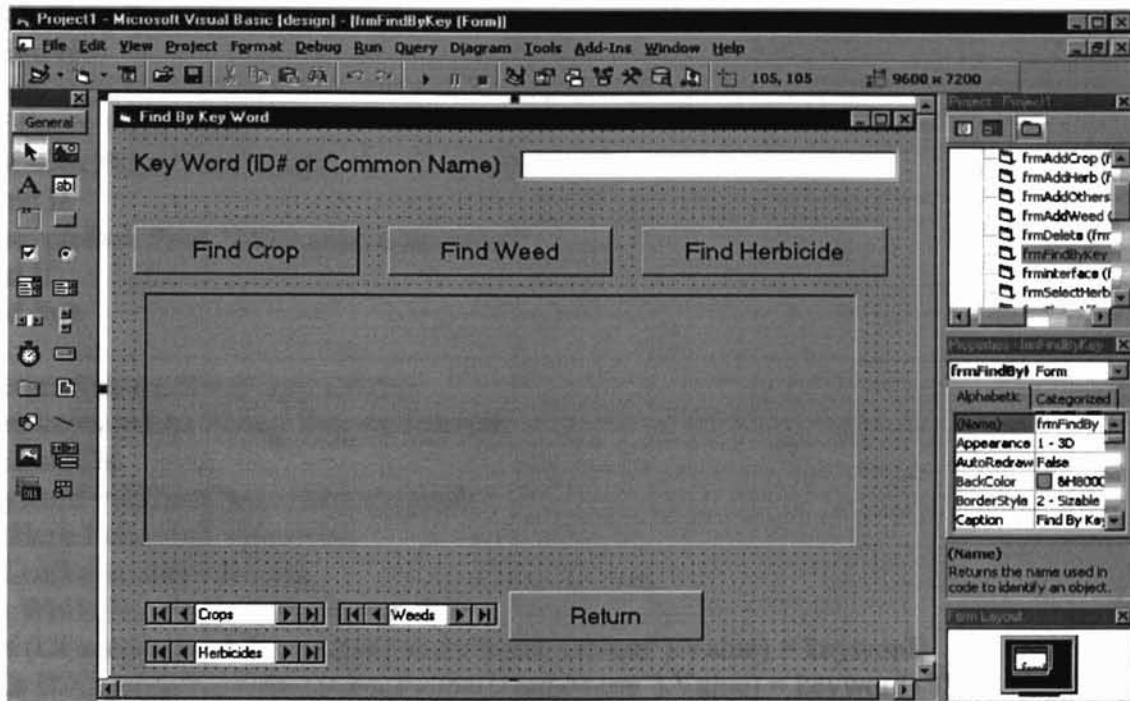
Else
    MsgBox "Must enter Rotational Crop name or ID ", , ""
End If
End Sub

Private Sub cmdDeleteWeed_Click()
Dim keyword As String, flag As Boolean
flag = False
keyword = UCase(InputBox("Weed Name or Weed ID", "Find Weed"))
If Len(keyword) > 0 Then
datWeed.Recordset.MoveFirst
Do While Not datWeed.Recordset.EOF
    If (UCase(datWeed.Recordset.Fields("WeedName").Value) = keyword) _
    Or (UCase(datWeed.Recordset.Fields("WeedIDNum").Value) = keyword) Then
        flag = True
        datWeed.Recordset.Delete
        MsgBox "record has been deleted", , ""
        datWeed.Recordset.MoveNext
    Else
        datWeed.Recordset.MoveNext
    End If
Loop
If flag = False Then
    MsgBox "No this kind of weed", , ""
End If
Else
    MsgBox "Must enter weed name or weed ID, , ""
End If
End Sub

Private Sub cmdReturn_Click()
frminterface.Show
frmDelete.Hide
End Sub

```

8. Form frmFindByKey and Code:



```
Private Sub cmdFindCrop_Click()
Dim keyword As String, flag As Boolean
picBox.Cls
keyword = UCase(txtSearchKey.Text)
datCrop.Recordset.MoveFirst
If Len(keyword) > 0 Then
Do While Not datCrop.Recordset.EOF
If (UCase(datCrop.Recordset.Fields("CropIDNum").Value) = keyword) _
Or (UCase(datCrop.Recordset.Fields("CropName").Value) = keyword) Then

flag = True
picBox.Print "Crop ID# : "; Tab(30); _
datCrop.Recordset.Fields("CropIDNum").Value
picBox.Print "Crop Name: "; Tab(30); _
datCrop.Recordset.Fields("CropName").Value
picBox.Print "Life Cycle: "; Tab(30); _
datCrop.Recordset.Fields("LifeCyl").Value
picBox.Print "Minimal Growth Stag :"; Tab(30); _
datCrop.Recordset.Fields("MinGroSta").Value
picBox.Print "Maximum Growth Stage: "; Tab(30); _
datCrop.Recordset.Fields("MaxGroSta").Value
picBox.Print "Reproduction: "; Tab(30); _
datCrop.Recordset.Fields("Reprod").Value
picBox.Print "Product Utilization: "; Tab(30); _
datCrop.Recordset.Fields("ProUtil").Value
```

```

datCrop.Recordset.MoveNext
Else
    datCrop.Recordset.MoveNext
End If
Loop
If flag = False Then
    picBox.Print "No Found"
End If
Else: picBox.Print "Must enter a keyword"
End If
End Sub

Private Sub cmdFindHerb_Click()
Dim keyword As String, flag As Boolean
picBox.Cls
keyword = UCase(txtSearchKey.Text)
datHerb.Recordset.MoveFirst
If Len(keyword) > 0 Then
Do While Not datHerb.Recordset.EOF
    If (UCase(datHerb.Recordset.Fields("HerbIDNum").Value) = keyword) _
    Or (UCase(datHerb.Recordset.Fields("HerbName").Value) = keyword) Then

        flag = True
        picBox.Print "Herbicide ID#: "; Tab(30); _
        datHerb.Recordset.Fields("HerbIDNum").Value
        picBox.Print "Herbicide Name: "; Tab(30); _
        datHerb.Recordset.Fields("HerbName").Value
        picBox.Print "Chemical Family: "; Tab(30); _
        datHerb.Recordset.Fields("ChFamily").Value
        picBox.Print "Formula: "; Tab(30); _
        datHerb.Recordset.Fields("Formula").Value
        picBox.Print "Mode of Action: "; Tab(30); _
        datHerb.Recordset.Fields("MOA").Value
        picBox.Print "Active Ingredient: "; Tab(30); _
        datHerb.Recordset.Fields("AI").Value
        picBox.Print "Application Rate: "; Tab(30); _
        datHerb.Recordset.Fields("ApplRate").Value
        picBox.Print "Temperature Restriction: "; _
        Tab(30); datHerb.Recordset.Fields("TepRestri").Value
        datHerb.Recordset.MoveNext
    Else
        datHerb.Recordset.MoveNext
    End If
Loop
If flag = False Then
    picBox.Print "No Found"

```

```

End If
Else: picBox.Print "Must enter a keyword"
End If
End Sub

Private Sub cmdFindWeed_Click()
Dim keyword As String, flag As Boolean
picBox.Cls
keyword = UCase(txtSearchKey.Text)
datWeed.Recordset.MoveFirst
If Len(keyword) > 0 Then
Do While Not datWeed.Recordset.EOF
If (UCase(datWeed.Recordset.Fields("WeedIDNum").Value) = keyword) _
Or (UCase(datWeed.Recordset.Fields("WeedName").Value) = keyword) Then

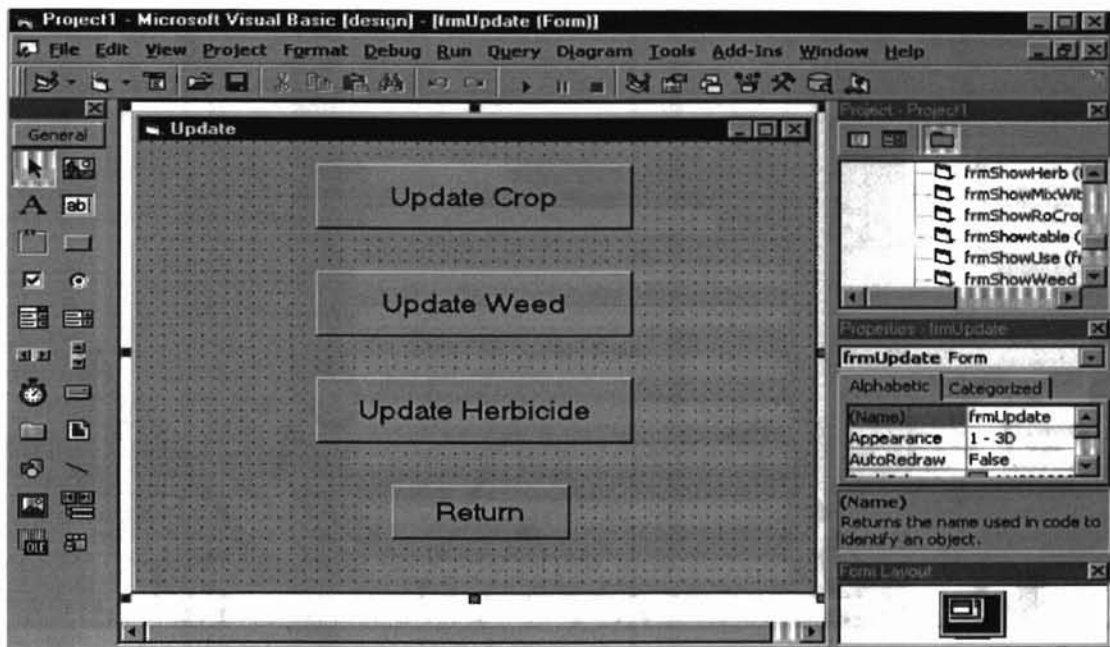
flag = True
picBox.Print "Weed ID# :      "; Tab(30); _
datWeed.Recordset.Fields("WeedIDNum").Value
picBox.Print "Weed Name:      "; Tab(30); _
datWeed.Recordset.Fields("WeedName").Value
picBox.Print "Life Cycle:      "; Tab(30); _
datWeed.Recordset.Fields("LifeCyl").Value
picBox.Print "Minimal Growth Stag :"; Tab(30); _
datWeed.Recordset.Fields("MinGroSta").Value
picBox.Print "Maximum Growth Stage: "; Tab(30); _
datWeed.Recordset.Fields("MaxGroSta").Value
picBox.Print "Reproduction:      "; Tab(30); _
datWeed.Recordset.Fields("Reprod").Value

datWeed.Recordset.MoveNext
Else
datWeed.Recordset.MoveNext
End If
Loop
If flag = False Then
picBox.Print "No Found"
End If
Else: picBox.Print "Must enter a keyword"
End If
End Sub

Private Sub cmdReturn_Click()
frmFindByKey.Hide
frminterface.Show
End Sub

```

9. Form frmUpdate and Code:



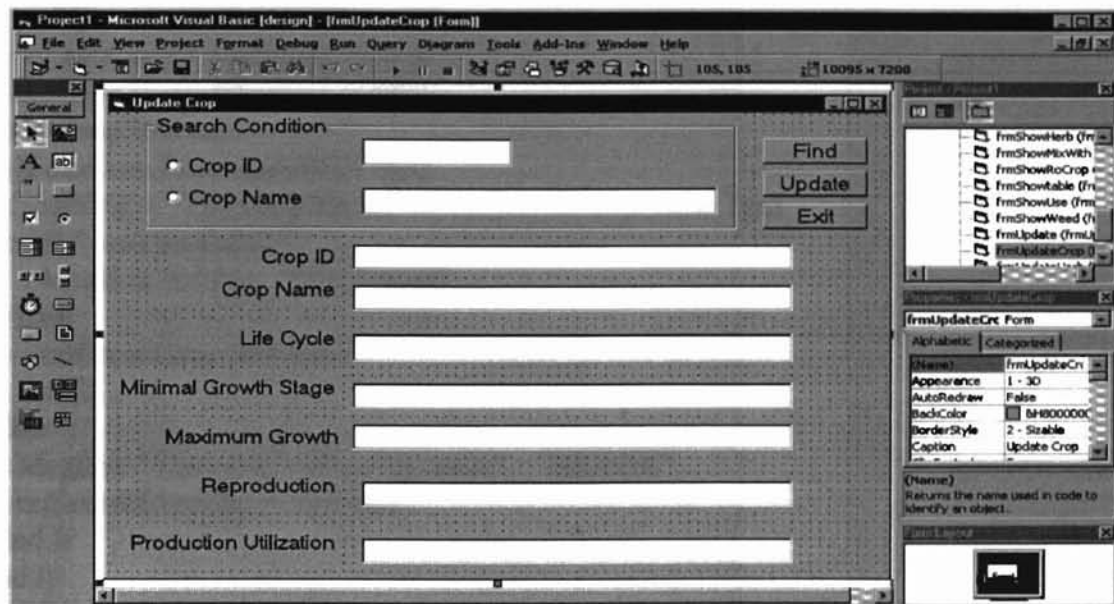
```
Private Sub cmdReturn_Click()
frminterface.Show
frmUpdate.Hide
End Sub
```

```
Private Sub cmdUpdateCrop_Click()
frmUpdateCrop.Show
frmUpdate.Hide
End Sub
```

```
Private Sub cmdUpdateHerb_Click()
frmUpdateHerb.Show
frmUpdate.Hide
End Sub
```

```
Private Sub cmdUpdateWeed_Click()
frmUpdateWeed.Show
frmUpdate.Hide
End Sub
```

10. Form frmUpdateCrop and Code:



Option Explicit

```
Private Sub cmdExit_Click()
```

```
'optCropID.SetFocus
```

```
frmUpdateCrop.Hide
```

```
frmUpdate.Show
```

```
End Sub
```

```
Private Sub cmdFind_Click()
```

```
Dim strSQL As String, strMsg As String
```

```
If optCropID = True Then
```

```
If Trim(txtSearchCropID.Text) <> "" Then
```

```
    strSQL = "SELECT * FROM Crop WHERE CropIDNum=" & """" & _  
        txtSearchCropID.Text & """"
```

```
    datCrop.RecordSource = strSQL
```

```
    datCrop.Refresh
```

```
    If datCrop.Recordset.EOF Then
```

```
        strMsg = "The Crop ID Number" & txtSearchCropID.Text _  
            & "isn't available"
```

```
        MsgBox strMsg, , "Not Found"
```

```
    Else
```

```
        txtCropName.SetFocus
```

```
    End If
```

```
Else
```

```
    MsgBox "You must enter the searching Crop ID number", , "ERROR"
```

```
    txtSearchCropID.SetFocus
```

```
End If
```

```

Else
    If Trim(txtSearchCropName.Text) <> "" Then
        strSQL = "SELECT * FROM Crop WHERE CropName=" & """" & _
            txtSearchCropName.Text & """"
        datCrop.RecordSource = strSQL
        datCrop.Refresh
        If datCrop.Recordset.BOF Then
            strMsg = "The Crop Name " & txtSearchCropName.Text _
                & " isn't available"
            MsgBox strMsg, , "Not Found"
        Else
            txtCropName.SetFocus
        End If
    Else
        MsgBox "You must enter crop name", , "ERROR"
        txtSearchCropName.SetFocus
    End If
End If
End Sub

```

```

Private Sub cmdUpdate_Click()
    'Update Crop table
    If txtSearchCropID.Text = "" And txtSearchCropName.Text = "" Then
        MsgBox "You must enter the search key", , "ERROR"
    Else
        If txtCropID.Text = "" Then
            MsgBox "You must find first, then update", , "ERROR"
        Else
            datCrop.Recordset.Edit
            datCrop.Recordset.Update
            MsgBox "Crop Table has been updated", , "SUCCESS!"
        End If
    End If
End Sub

```

```

Private Sub Form_Activate()
    'clear the text boxes
    optCropID.SetFocus
    txtSearchCropID.Text = ""
    txtSearchCropName.Text = ""
    txtCropID.Text = ""
    txtCropName.Text = ""
    txtLifeCyl.Text = ""
    txtMinGroSta.Text = ""
    txtMaxGroSta.Text = ""
    txtReprod.Text = ""

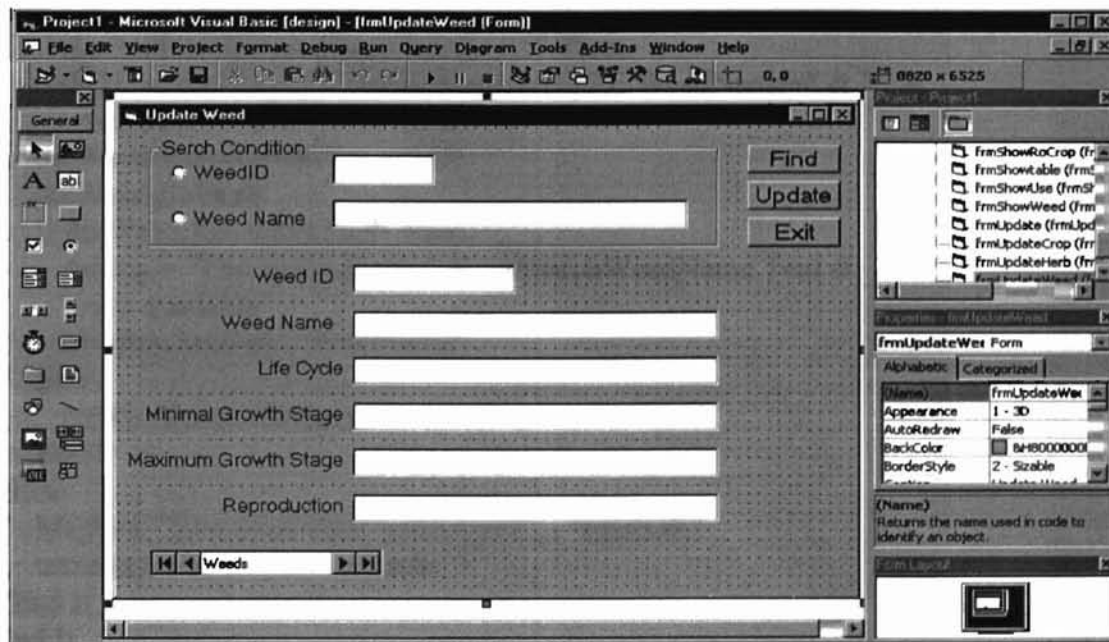
```

```
txtProUtil.Text = ""
End Sub
```

```
Private Sub optCropID_Click()
'Select Crop ID as Searching Condition
txtSearchCropID.Text = ""
txtSearchCropID.Visible = True
txtSearchCropName.Visible = False
End Sub
```

```
Private Sub optCropName_Click()
'Select Crop Name As Searching Condition
txtSearchCropName.Text = ""
txtSearchCropID.Visible = False
txtSearchCropName.Visible = True
End Sub
```

11. Form frmUpdateWeed and Code:



```
Option Explicit
Private Sub cmdExit_Click()
'optWeedID.SetFocus
frmUpdateWeed.Hide
frmUpdate.Show
```

End Sub

Private Sub cmdFind_Click()

Dim strSQL As String, strMsg As String

If optWeedID = True Then

If Trim(txtSearchWeedID.Text) <> "" Then

strSQL = "SELECT * FROM Weed WHERE WeedIDNum=" & """" & _
txtSearchWeedID.Text & """"

datWeed.RecordSource = strSQL

datWeed.Refresh

If datWeed.Recordset.BOF Then

strMsg = "The Weed ID Number" & txtSearchWeedID.Text & _
& "isn't available"

MsgBox strMsg, , "Not Found"

Else

txtWeedName.SetFocus

End If

Else

MsgBox "You must enter the searching weed ID number", , "ERROR"

txtSearchWeedID.SetFocus

End If

Else

If Trim(txtSearchWeedName.Text) <> "" Then

strSQL = "SELECT * FROM Weed WHERE WeedName=" & """" & _
txtSearchWeedName.Text & """"

datWeed.RecordSource = strSQL

datWeed.Refresh

If datWeed.Recordset.BOF Then

strMsg = "The Weed Name " & txtSearchWeedName.Text & _
"isn't available"

MsgBox strMsg, , "Not Found"

Else

txtWeedName.SetFocus

End If

Else

MsgBox "You must enter weed name", , "ERROR"

txtSearchWeedName.SetFocus

End If

End If

End Sub

Private Sub cmdUpdate_Click()

'Update Weed table

If txtSearchWeedID.Text = "" And txtSearchWeedName.Text = "" Then

MsgBox "You must enter the search key", , "ERROR"

Else

```

If txtWeedID.Text = "" Then
MsgBox "You must find first, then update", , "ERROR"
Else
datWeed.Recordset.Edit
datWeed.Recordset.Update
MsgBox "Weed Table has been updated", , "SUCCESS!"
End If
End If
End Sub

```

```

Private Sub Form_Activate()
'clear the text boxes
optWeedID.SetFocus
txtSearchWeedID = ""
txtSearchWeedName = ""
txtWeedID = ""
txtWeedName = ""
txtLifeCyl = ""
txtMinGroSta = ""
txtMaxGroSta = ""
txtReprod = ""
End Sub

```

```

Private Sub optWeedID_Click()
'Select Weed ID as Searching Condition
txtSearchWeedID.Text = ""
txtSearchWeedID.Visible = True
txtSearchWeedName.Visible = False
End Sub

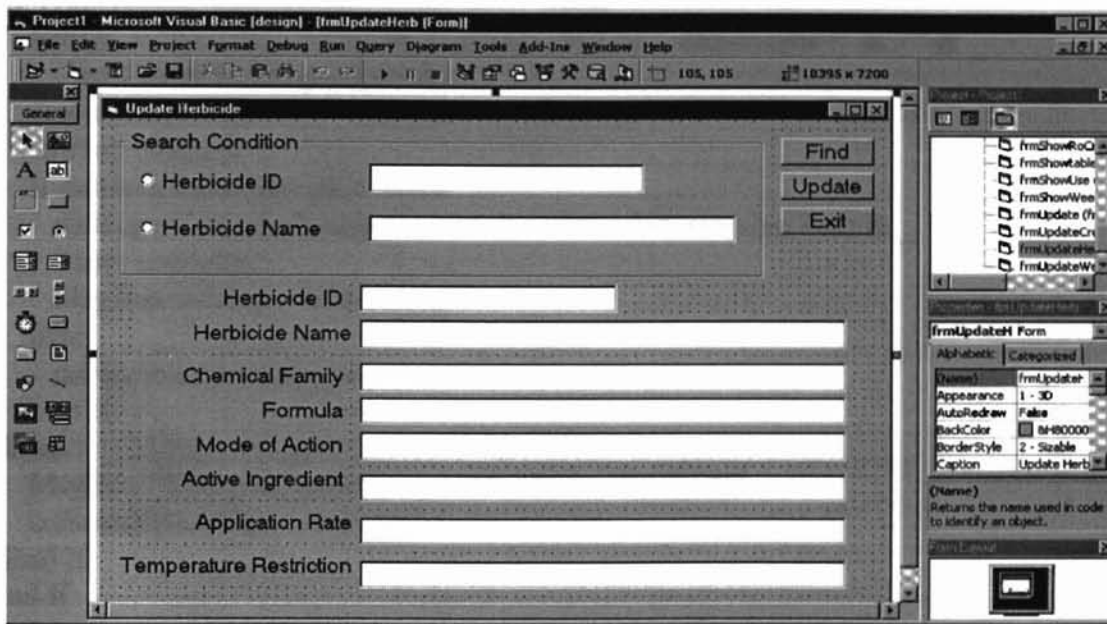
```

```

Private Sub optWeedName_Click()
'Select Weed Name As Searching Condition
txtSearchWeedName.Text = ""
txtSearchWeedID.Visible = False
txtSearchWeedName.Visible = True
End Sub

```

12. Form frmUpdateHerb and Code:



Option Explicit

```
Private Sub cmdExit_Click()
'frmUpdateHerb.Hide
frmUpdate.Show
End Sub
```

```
Private Sub cmdFind_Click()
Dim strSQL As String, strMsg As String
If optHerbID = True Then
If Trim(txtSearchHerbID.Text) <> "" Then
strSQL = "SELECT * FROM Herbicide WHERE HerbIDNum=" & """" & _
txtSearchHerbID.Text & """"
datHerb.RecordSource = strSQL
datHerb.Refresh
If datHerb.Recordset.EOF Then
strMsg = "The Herbicide ID Number " & txtSearchHerbID.Text & _
" isn't available"
MsgBox strMsg, , "Not Found"
Else
txtHerbName.SetFocus
End If
Else
MsgBox "You must enter the searching Herbicide ID number", , "ERROR"
txtSearchHerbID.SetFocus
End If
```

```

Else
    If Trim(txtSearchHerbName.Text) <> "" Then
        strSQL = "SELECT * FROM Herbicide WHERE HerbName=" & """" & _
            txtSearchHerbName.Text & """"
        datHerb.RecordSource = strSQL
        datHerb.Refresh
        If datHerb.Recordset.BOF Then
            strMsg = "The Herbicide Name " & txtSearchHerbName.Text & _
                " isn't available"
            MsgBox strMsg, , "Not Found"
        Else
            txtHerbName.SetFocus
        End If
    Else
        MsgBox "You must enter herbicide name", , "ERROR"
        txtSearchHerbName.SetFocus
    End If
End If
End Sub

```

```

Private Sub cmdUpdate_Click()
    'Update Herbicide table
    If txtSearchHerbID.Text = "" And txtSearchHerbName.Text = "" Then
        MsgBox "You must enter the search key", , "ERROR"
    Else
        If txtHerbID.Text = "" Then
            MsgBox "You must find first, then update", , "ERROR"
        Else
            datHerb.Recordset.Edit
            datHerb.Recordset.Update
            MsgBox "Herbicide Table has been updated", , "SUCCESS!"
        End If
    End If
End Sub

```

```

Private Sub Form_Activate()
    'clear the text boxes
    optHerbID.SetFocus
    txtSearchHerbID.Text = ""
    txtSearchHerbName.Text = ""
    txtHerbID.Text = ""
    txtHerbName.Text = ""
    txtChFamily.Text = ""
    txtFormula.Text = ""
    txtMOA.Text = ""
    txtAI.Text = ""

```



```
txtApplRate.Text = ""
txtTepRestri.Text = ""
End Sub
```

```
Private Sub optHerbID_Click()
'Select Herbicide ID as Searching Condition
txtSearchHerbID.Text = ""
txtSearchHerbID.Visible = True
txtSearchHerbName.Visible = False
End Sub
```

```
Private Sub optHerbName_Click()
'Select Herbicide Name As Searching Condition
txtSearchHerbName.Text = ""
txtSearchHerbID.Visible = False
txtSearchHerbName.Visible = True
End Sub
```

13. Form frmShowtables and Code:



```
Private Sub cmdReturn_Click()
frmShowtable.Hide
frminterface.Show
End Sub
```

```
Private Sub cmdShowAffect_Click()
frmShowAffect.Show
frmShowtable.Hide
```

End Sub

```
Private Sub cmdShowCompaProd_Click()  
frmShowCompaProd.Show  
frmShowtable.Hide  
End Sub
```

```
Private Sub cmdShowControl_Click()  
frmShowControl.Show  
frmShowtable.Hide  
End Sub
```

```
Private Sub cmdShowCrop_Click()  
frmShowCrop.Show  
frmShowtable.Hide  
End Sub
```

```
Private Sub cmdShowHerb_Click()  
frmShowHerb.Show  
frmShowtable.Hide  
End Sub
```

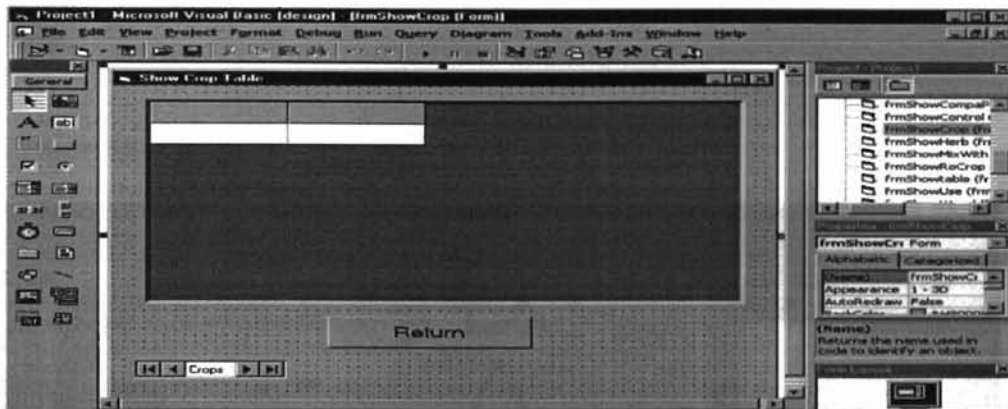
```
Private Sub cmdShowMixWith_Click()  
frmShowMixWith.Show  
frmShowtable.Hide  
End Sub
```

```
Private Sub cmdShowRoCrop_Click()  
frmShowRoCrop.Show  
frmShowtable.Hide  
End Sub
```

```
Private Sub cmdShowUse_Click()  
frmShowUse.Show  
frmShowtable.Hide  
End Sub
```

```
Private Sub cmdShowWeed_Click()  
frmShowWeed.Show  
frmShowtable.Hide  
End Sub
```

14. Form frmShowCrop and Code:



```
Private Sub cmdReturn_Click()
frmShowCrop.Hide
frmShowtable.Show
End Sub
```

```
Private Sub msgCrop_Click()
msgCrop.ColWidth(2) = 2000
msgCrop.ColWidth(5) = 2000
msgCrop.ColWidth(6) = 2000
End Sub
```

15. Form frmShowWeed and Code:



```
Private Sub cmdReturn_Click()
frmShowWeed.Hide
frmShowtable.Show
End Sub
```

```

Private Sub msgWeed_Click()
msgWeed.ColWidth(1) = 2000
msgWeed.ColWidth(2) = 2000
End Sub

```

16. Form frmShowHerb and Code:



```

Private Sub cmdReturn_Click()
frmShowHerb.Hide
frmShowtable.Show
End Sub

```

```

Private Sub msgHerb_Click()
msgHerb.ColWidth(1) = 2000
msgHerb.ColWidth(2) = 2500
msgHerb.ColWidth(4) = 2500
End Sub

```

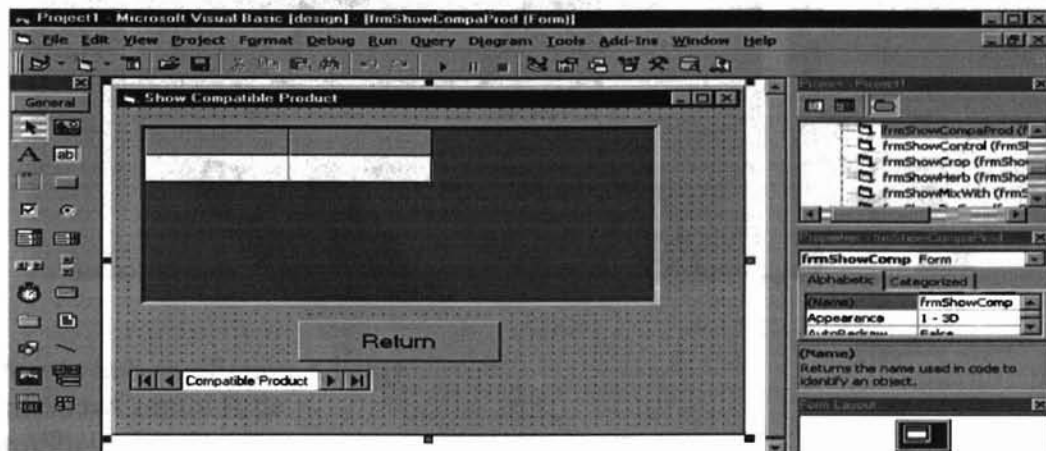
17. Form frmShowRoCrop and Code:



```
Private Sub cmdReturn_Click()
frmShowRoCrop.Hide
frmShowtable.Show
End Sub
```

```
Private Sub msgRoCrop_Click()
msgRoCrop.ColWidth(0) = 2000
msgRoCrop.ColWidth(1) = 2000
End Sub
```

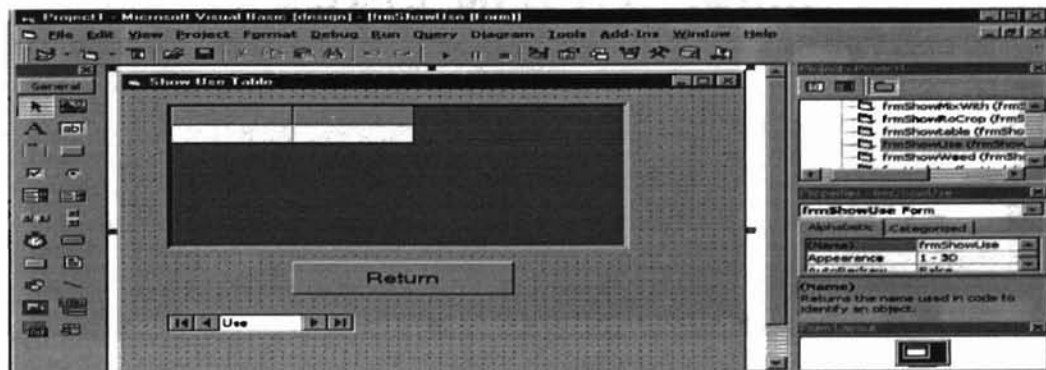
18. Form frmShowCompaProd and Code:



```
Private Sub cmdReturn_Click()
frmShowCompaProd.Hide
frmShowtable.Show
End Sub
```

```
Private Sub msgCompaProd_Click()
msgCompaProd.ColWidth(1) = 2000
End Sub
```

19. Form frmShowUse and Code:

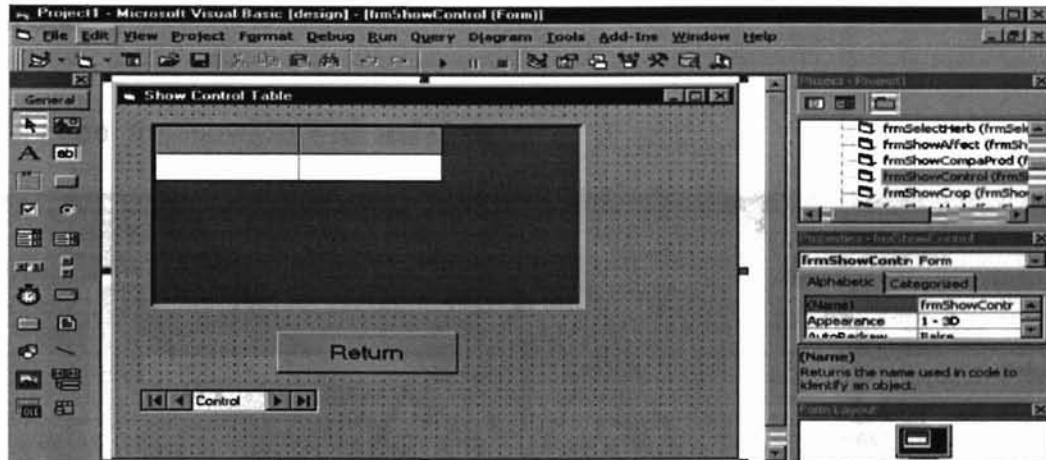


```

Private Sub cmdReturn_Click()
frmShowUse.Hide
frmShowtable.Show
End Sub

```

20. Form frmShowControl and Code:



```

Private Sub cmdReturn_Click()
frmShowControl.Hide
frmShowtable.Show
End Sub

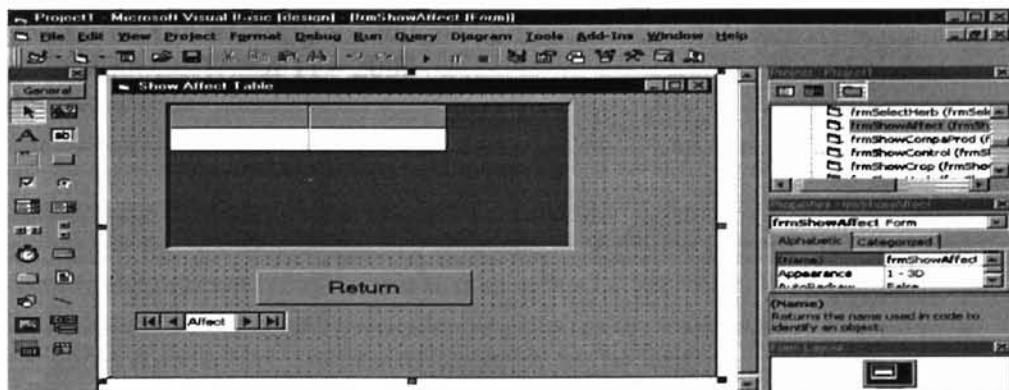
```

```

Private Sub msgControl_Click()
msgControl.ColWidth(0) = 2000
msgControl.ColWidth(1) = 2000
End Sub

```

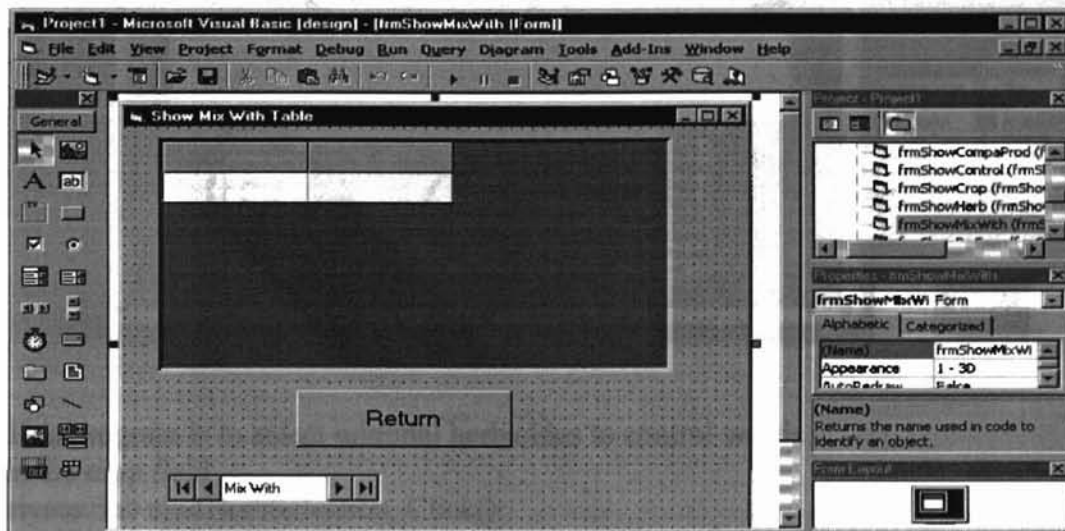
21. Form frmShowAffect and Code:



```
Private Sub cmdReturn_Click()
frmShowAffect.Hide
frmShowtable.Show
End Sub
```

```
Private Sub msgAffect_Click()
msgAffect.ColWidth(0) = 2000
msgAffect.ColWidth(1) = 2000
End Sub
```

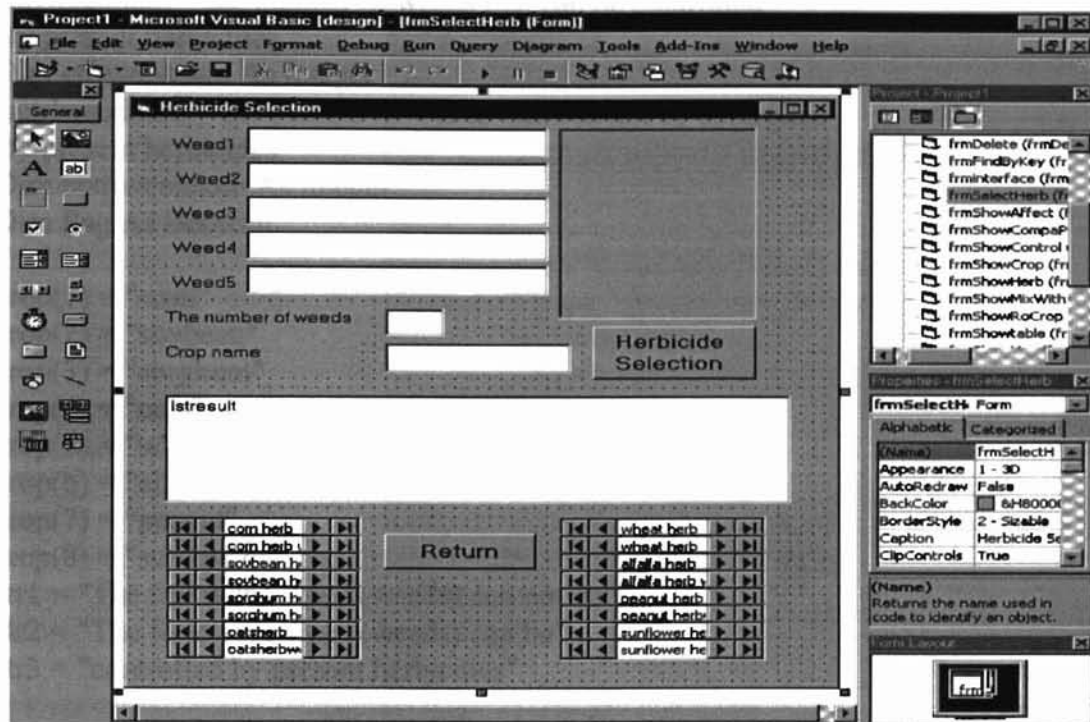
22. Form frmShowMixWith and Code:



```
Private Sub cmdReturn_Click()
frmShowMixWith.Hide
frmShowtable.Show
End Sub
```

```
Private Sub msgMixWith_Click()
msgMixWith.ColWidth(0) = 2000
msgMixWith.ColWidth(1) = 2000
End Sub
```

23. Form frmSelectHerb and Code:



' This program is to select minimal herbicides to control weeds
' in the crop field

Private Sub cmdHerbSelection_Click()

'weednum() define the number of weeds that will be probably controlled

Dim weednum(1 To 5) As String

'crop() define the number of crops

Dim crop(1 To 8) As String

Dim str1 As String, str2 As String, str3 As String, str4 As String

Dim i As Integer, j As Integer, k As Integer

'wnum represents the number of the weeds the farmers want to control

Dim wnum As Integer

'q represents the number of herbicides that are being used in different
'crop fields, respectively

Dim q As Integer

Dim p As Integer

'count record the number of weeds in the loop

Dim count As Integer

'maxcount = the maximum number of weeds that can be controlled
'by one herbicide among all herbicides

Dim maxcount As Integer

'hbposition records the position of herbicide in the loop

Dim hbposition As Integer

'finalhbposition = the selected,desired herbicide's position.


```

'this kind of herbicide can control maximum weeds
Dim finalhbposition As Integer
'ctrlweednum = the number of weeds that can be controlled
'by present herbicides
Dim ctrlweednum As Integer
'nctrlweednum = the number of weeds that can not be controlled
'by present herbicides
Dim nctrlweednum As Integer
Dim flag As Boolean

crop(1) = "corn"
crop(2) = "soybean"
crop(3) = "sorghum"
crop(4) = "oats"
crop(5) = "wheat"
crop(6) = "alfalfa"
crop(7) = "peanut"
crop(8) = "sunflower"
str1 = "The following weed (weeds) can not be"
str2 = "The following weed (weeds) can be"
str3 = "controlled by present herbicides"

Rem initialization
ctrlweednum = 0
nctrlweednum = 0
count = 0
maxcount = 0
hbposition = 0
finalhbposition = 0
flag = False
picbox.Cls
lstresult.Clear

Rem assignment
wnum = Val(txtweednum.Text)
weednum(1) = txtweed1.Text
weednum(2) = txtweed2.Text
weednum(3) = txtweed3.Text
weednum(4) = txtweed4.Text
weednum(5) = txtweed5.Text
If txtweednum.Text = "" Or txtweed1.Text = "" _
Or txtcrop.Text = "" Then
MsgBox "You must input the weeds that will be controlled," _
+ " The number of weeds and crop name at the same time", , ""

Else

```

```

'weed() array store the number of weeds that farmers want to control
ReDim weed(1 To wnum) As Integer

'determine if weeds appear in the corn field

If crop(1) = txtcrop.Text Then
datcornherb.Recordset.MoveLast 'needed to set value of RecordCount
datcornherb.Recordset.MoveFirst
q = datcornherb.Recordset.RecordCount
ReDim herblist(1 To q) As String 'dynamically define herbicide array
ReDim matrix(1 To q, 1 To wnum) As Integer 'dynamically define two-way
                                     'array, used for recording
                                     'which weed can be controlld
                                     'by which herbicide

'set up herbicide list that can be used in different crop field
For j = 1 To q
    herblist(j) = datcornherb.Recordset.Fields("HerbName").Value
    datcornherb.Recordset.MoveNext
Next j

'set up weed list that the farmers want to control
For i = 1 To wnum
    weed(i) = weednum(i)
Next i

'compare and determine the relationship between herbicides and weeds
'and set up matrix
For p = 1 To q
    For k = 1 To wnum
datcornherb.Recordset.MoveLast
datcornherb.Recordset.MoveFirst
datcornherbweed.Recordset.MoveLast
datcornherbweed.Recordset.MoveFirst
For i = 1 To datcornherbweed.Recordset.RecordCount
If (UpperCase(datcornherbweed.Recordset.Fields("HerbName").Value) = (UpperCase _
(herblist(p)))) And (UpperCase(datcornherbweed.Recordset.Fields _
("WeedName").Value) = (UpperCase(weednum(k)))) Then
    matrix(p, k) = 1
End If
datcornherbweed.Recordset.MoveNext
Next i
Next k
datcornherb.Recordset.MoveNext
Next p

```

```

'print out matrix
For i = 1 To q
    For j = 1 To wnum
        picbox.Print matrix(i, j);
    Next j
picbox.Print
Next i
End If

'determine if weeds appear in the soybean field
If crop(2) = txtcrop.Text Then

    datsoyherb.Recordset.MoveLast 'needed to set value of RecordCount
    datsoyherb.Recordset.MoveFirst
    q = datsoyherb.Recordset.RecordCount
    ReDim herblist(1 To q) As String 'dynamically define herbicide array
    ReDim matrix(1 To q, 1 To wnum) As Integer 'dynamically define two-way
        'array, used for recording
        'which weed can be controlld
        'by which herbicide

    'set up herbicide list that can be used in different crop field
    For j = 1 To q
        herblist(j) = datsoyherb.Recordset.Fields("HerbName").Value
    Next j
    datsoyherb.Recordset.MoveNext
Next j
'set up weed list that the farmers want to control
For i = 1 To wnum
    weed(i) = weednum(i)
Next i
'compare and determine the relationship between herbicides and weeds
'and set up matrix
For p = 1 To q
    For k = 1 To wnum
        datsoyherb.Recordset.MoveLast
        datsoyherb.Recordset.MoveFirst

        datsoyherbweed.Recordset.MoveLast
        datsoyherbweed.Recordset.MoveFirst

        For i = 1 To datsoyherbweed.Recordset.RecordCount
            If (UCase(datsoyherbweed.Recordset.Fields("HerbName").Value) = (UCase _
                (herblist(p)))) And (UCase(datsoyherbweed.Recordset.Fields _
                ("WeedName").Value) = (UCase(weed(k)))) Then
                matrix(p, k) = 1
            End If
        Next i
    Next k
Next p

```

```

    datsoyherbweed.Recordset.MoveNext
Next i
Next k

    datsoyherb.Recordset.MoveNext
Next p

'print out matrix
For i = 1 To q
    For j = 1 To wnum
        picbox.Print matrix(i, j);
    Next j
    picbox.Print
Next i
End If

'determine if weeds appear in the sorghum field
If crop(3) = txtcrop.Text Then
    datsorghumherb.Recordset.MoveLast
    datsorghumherb.Recordset.MoveFirst
    q = datsorghumherb.Recordset.RecordCount
    ReDim herblist(1 To q) As String 'dynamically define herbicide array
    ReDim matrix(1 To q, 1 To wnum) As Integer 'dynamically define two-way
        'array, used for recording
        'which weed can be controlld
        'by which herbicide

'set up herbicide list that can be used in different crop field
For j = 1 To q
    herblist(j) = datsorghumherb.Recordset.Fields("HerbName").Value
    datsorghumherb.Recordset.MoveNext
Next j
'set up weed list that the farmers want to control
For i = 1 To wnum
    weed(i) = weednum(i)
Next i
'compare and determine the relationship between herbicides and weeds
'and set up matrix
For p = 1 To q
    For k = 1 To wnum
        datsorghumherb.Recordset.MoveLast
        datsorghumherb.Recordset.MoveFirst

datsorghumherbweed.Recordset.MoveLast
datsorghumherbweed.Recordset.MoveFirst

```

```

For i = 1 To datsorghumherbweed.Recordset.RecordCount
If (UCase(datsorghumherbweed.Recordset.Fields("HerbName").Value) = _
(UCase(herblist(p)))) And (UCase(datsorghumherbweed.Recordset.Fields _
("WeedName").Value) = (UCase(weed(k)))) Then

    matrix(p, k) = 1

End If

    datsorghumherbweed.Recordset.MoveNext

Next i

Next k
datsorghumherb.Recordset.MoveNext
Next p
'print out matrix
For i = 1 To q
    For j = 1 To wnum
        picbox.Print matrix(i, j);
    Next j
    picbox.Print
Next i
End If

'determine if weeds appear in the oats field
If crop(4) = txtcrop.Text Then
datoatsherb.Recordset.MoveLast
datoatsherb.Recordset.MoveFirst
q = datoatsherb.Recordset.RecordCount
ReDim herblist(1 To q) As String 'dynamically define herbicide array
ReDim matrix(1 To q, 1 To wnum) As Integer 'dynamically define two-way
                                'array, used for recording
                                'which weed can be controlld
                                'by which herbicide

'set up herbicide list that can be used in different crop field
For j = 1 To q
herblist(j) = datoatsherb.Recordset.Fields("HerbName").Value
datoatsherb.Recordset.MoveNext
Next j
'set up weed list that the farmers want to control
For i = 1 To wnum

```

```

    weed(i) = weednum(i)
Next i
'compare and determine the relationship between herbicides and weeds
'and set up matrix
For p = 1 To q
    For k = 1 To wnum
datoatsherb.Recordset.MoveLast
datoatsherb.Recordset.MoveFirst
datoatsherbweed.Recordset.MoveLast
datoatsherbweed.Recordset.MoveFirst

    For i = 1 To datoatsherbweed.Recordset.RecordCount
    If (UCase(datoatsherbweed.Recordset.Fields("HerbName").Value) = _
(UCase(herblist(p)))) And (UCase(datoatsherbweed.Recordset.Fields _
("WeedName").Value) = (UCase(weed(k)))) Then

        matrix(p, k) = 1

    End If
    datoatsherbweed.Recordset.MoveNext
    Next i
    Next k
    datoatsherb.Recordset.MoveNext
    Next p
    'print out matrix
    For i = 1 To q
        For j = 1 To wnum
            picbox.Print matrix(i, j);
        Next j
        picbox.Print
    Next i
End If

'determine if weeds appear in the wheat field
If crop(5) = txtcrop.Text Then
datwheatherb.Recordset.MoveLast
datwheatherb.Recordset.MoveFirst
q = datwheatherb.Recordset.RecordCount
ReDim herblist(1 To q) As String 'dynamically define herbicide array
ReDim matrix(1 To q, 1 To wnum) As Integer 'dynamically define two-way
        'array, used for recording
        'which weed can be controlld
        'by which herbicide

'set up herbicide list that can be used in different crop field
For j = 1 To q

```

```

herblist(j) = datwheatherb.Recordset.Fields("HerbName").Value
datwheatherb.Recordset.MoveNext
Next j
'set up weed list that the farmers want to control
For i = 1 To wnum
    weed(i) = weednum(i)
Next i
'compare and determine the relationship between herbicides and weeds
'and set up matrix
For p = 1 To q
    For k = 1 To wnum
datwheatherb.Recordset.MoveLast
datwheatherb.Recordset.MoveFirst

datwheatherbweed.Recordset.MoveLast
datwheatherbweed.Recordset.MoveFirst

For i = 1 To datwheatherbweed.Recordset.RecordCount
If (UCase(datwheatherbweed.Recordset.Fields("HerbName").Value) = _
(UCase(herblist(p)))) And (UCase(datwheatherbweed.Recordset.Fields _
("WeedName").Value) = (UCase(weed(k)))) Then

    matrix(p, k) = 1

End If
    datwheatherbweed.Recordset.MoveNext
Next i
Next k
    datwheatherb.Recordset.MoveNext
Next p
'print out matrix
For i = 1 To q
    For j = 1 To wnum
        picbox.Print matrix(i, j);
    Next j
    picbox.Print
Next i
End If

'determine if weeds appear in the alfalfa field
If crop(6) = txtcrop.Text Then
dataalfaherb.Recordset.MoveLast
dataalfaherb.Recordset.MoveFirst
q = dataalfaherb.Recordset.RecordCount
ReDim herblist(1 To q) As String 'dynamically define herbicide array
ReDim matrix(1 To q, 1 To wnum) As Integer 'dynamically define two-way

```

```

                                'array, used for recording
                                'which weed can be controlld
                                'by which herbicide

'set up herbicide list that can be used in different crop field
For j = 1 To q
herblist(j) = datalfafaherb.Recordset.Fields("HerbName").Value
datalfafaherb.Recordset.MoveNext
Next j
'set up weed list that the farmers want to control
For i = 1 To wnum
    weed(i) = weednum(i)
Next i
'compare and determine the relationship between herbicides and weeds
'and set up matrix
For p = 1 To q
    For k = 1 To wnum
datalfafaherb.Recordset.MoveLast
datalfafaherb.Recordset.MoveFirst

datalfafaherbweed.Recordset.MoveLast
datalfafaherbweed.Recordset.MoveFirst

For i = 1 To datalfafaherbweed.Recordset.RecordCount
If (UCase(datalfafaherbweed.Recordset.Fields("HerbName").Value) = _
(UCase(herblist(p)))) And (UCase(datalfafaherbweed.Recordset.Fields _
("WeedName").Value) = (UCase(weed(k)))) Then

    matrix(p, k) = 1

End If

    datalfafaherbweed.Recordset.MoveNext
Next i
Next k
datalfafaherb.Recordset.MoveNext
Next p

'print out matrix
For i = 1 To q
    For j = 1 To wnum
        picbox.Print matrix(i, j);
    Next j
    picbox.Print
Next i
End If

```



```

'determine if weeds appear in the peanut field
If crop(7) = txtcrop.Text Then
datpeanutherb.Recordset.MoveLast
datpeanutherb.Recordset.MoveFirst
q = datpeanutherb.Recordset.RecordCount
ReDim herblist(1 To q) As String 'dynamically define herbicide array
ReDim matrix(1 To q, 1 To wnum) As Integer 'dynamically define two-way
                                         'array, used for recording
                                         'which weed can be controlld
                                         'by which herbicide

'set up herbicide list that can be used in different crop field
For j = 1 To q
herblist(j) = datpeanutherb.Recordset.Fields("HerbName").Value
datpeanutherb.Recordset.MoveNext
Next j
'set up weed list that the farmers want to control
For i = 1 To wnum
    weed(i) = weednum(i)
Next i
'compare and determine the relationship between herbicides and weeds
'and set up matrix
For p = 1 To q
    For k = 1 To wnum
datpeanutherb.Recordset.MoveLast
datpeanutherb.Recordset.MoveFirst

datpeanutherbweed.Recordset.MoveLast
datpeanutherbweed.Recordset.MoveFirst

For i = 1 To datpeanutherbweed.Recordset.RecordCount
If (UCase(datpeanutherbweed.Recordset.Fields("HerbName").Value) = _
(UCase(herblist(p)))) And (UCase(datpeanutherbweed.Recordset.Fields _
("WeedName").Value) = (UCase(weed(k)))) Then

    matrix(p, k) = 1

End If
datpeanutherbweed.Recordset.MoveNext

Next i
Next k
datpeanutherb.Recordset.MoveNext
Next p
'print out matrix

```

```

For i = 1 To q
  For j = 1 To wnum
    picbox.Print matrix(i, j);
  Next j
  picbox.Print
Next i
End If

'determine if weeds appear in the sunflower field
If crop(8) = txtcrop.Text Then
  datsunflowerherb.Recordset.MoveLast
  datsunflowerherb.Recordset.MoveFirst
  q = datsunflowerherb.Recordset.RecordCount
  ReDim herblist(1 To q) As String 'dynamically define herbicide array
  ReDim matrix(1 To q, 1 To wnum) As Integer 'dynamically define two-way
      'array, used for recording
      'which weed can be controlld
      'by which herbicide

'set up herbicide list that can be used in different crop field
For j = 1 To q
  herblist(j) = datsunflowerherb.Recordset.Fields("HerbName").Value
  datsunflowerherb.Recordset.MoveNext
Next j
'set up weed list that the farmers want to control
For i = 1 To wnum
  weed(i) = weednum(i)
Next i
'compare and determine the relationship between herbicides and weeds
'and set up matrix
For p = 1 To q
  For k = 1 To wnum
    datsunflowerherb.Recordset.MoveLast
    datsunflowerherb.Recordset.MoveFirst

    datsunflowerherbweed.Recordset.MoveLast
    datsunflowerherbweed.Recordset.MoveFirst

    For i = 1 To datsunflowerherbweed.Recordset.RecordCount
      If (UCase(datsunflowerherbweed.Recordset.Fields("HerbName").Value) = _
        (UCase(herblist(p)))) And (UCase(datsunflowerherbweed.Recordset.Fields _
        ("WeedName").Value) = (UCase(weed(k)))) Then

        matrix(p, k) = 1
      End If
      datsunflowerherbweed.Recordset.MoveNext
    
```

```

Next i
Next k
    datsunflowerherb.Recordset.MoveNext
Next p
'print out matrix
For i = 1 To q
    For j = 1 To wnum
        picbox.Print matrix(i, j);
    Next j
    picbox.Print
Next i
End If

```

```

'ctrlweed() store the weeds that can be
'controlledby present herbicides
ReDim ctrlweed(1 To wnum) As Integer

```

```

'nctrlweed() store the weeds that can't be
'controlled by present herbicides
ReDim nctrlweed(1 To wnum) As Integer

```

```

Call seekmaxcount(wnum, q, matrix(), herblist(), weed(), maxcount, _
    finalhbposition, ctrlweed(), nctrlweed(), _
    ctrlweednum, nctrlweednum, hbposition, count, _
    str1, str2, str3)

```

```

Select Case maxcount

```

```

    Case 0

```

```

        Call zeroweed

```

```

    Case 1

```

```

        Call oneweeds(wnum, q, matrix(), herblist(), weed(), maxcount, _
            finalhbposition, ctrlweed(), nctrlweed(), _
            ctrlweednum, nctrlweednum)

```

```

    Case 2

```

```

        Call twoweeds(wnum, q, matrix(), herblist(), weed(), maxcount, _
            finalhbposition, ctrlweed(), nctrlweed(), _
            ctrlweednum, nctrlweednum)

```

```

    Case 3

```

```

        Call threeweeds(wnum, q, matrix(), herblist(), weed(), _
            maxcount, finalhbposition, ctrlweed(), _
            nctrlweed(), ctrlweednum, nctrlweednum)

```

Case 4

```
Call fourweed(wnum, q, matrix(), herblist(), weed(), maxcount, _  
finalhbposition, ctrlweed(), nctrlweed(), _  
ctrlweednum, nctrlweednum)
```

Case 5

```
Call fiveweed(wnum, q, matrix(), herblist(), weed(), maxcount, _  
finalhbposition, ctrlweed(), nctrlweed(), ctrlweednum, _  
nctrlweednum)
```

End Select

End If

End Sub

Public Sub zeroweed()

```
lstresult.AddItem " the present herbicides can not control any" _  
+ " one of those weeds."
```

End Sub

```
Public Sub oneweeds(wnum As Integer, q As Integer, matrix() _  
As Integer, herblist() As String, weed() As String, _  
maxcount As Integer, finalhbposition As Integer, _  
ctrlweed() As Integer, nctrlweed() As Integer, _  
ctrlweednum As Integer, nctrlweednum As Integer)
```

'The maximum number of the weeds controlled by individual herbicide is one

```
Dim weedname As String
```

```
num = ctrlweednum - maxcount
```

```
If maxcount = 1 And ctrlweednum = 1 Then
```

```
For j = 1 To wnum
```

```
    If ctrlweed(j) = 1 Then
```

```
        weedname = weed(j)
```

```
    End If
```

```
Next j
```

```
lstresult.AddItem herblist(finalhbposition) + " can control " + weedname
```

```
lstresult.AddItem ""
```

```
End If
```

```
If maxcount = 1 And ctrlweednum > 1 Then
```

```
For i = 1 To wnum
```

```
    For j = 1 To q
```

```
        If matrix(j, i) = 1 And ctrlweed(i) = 1 Then
```

```
            lstresult.AddItem herblist(j) + " can control " + weed(i)
```

```
        End If
```

```
    Next j
```

```
Next i
```

```
End If
End Sub
```

```
Public Sub twoweed(wnum As Integer, q As Integer, matrix() As Integer, _
    herblist() As String, weed() As String, maxcount As _
    Integer, finalhbposition As Integer, ctrlweed() As _
    Integer, nctrlweed() As Integer, ctrlweednum As _
    Integer, nctrlweednum As Integer)

' Tme maximum number of the weeds controlled by individul herbicide is two
Dim weedname As String
Dim num As Integer, hbpos As Integer, count1 As Integer, bigcount As Integer, lastpos
As Integer
Dim twoarr(1 To 5) As Integer
Dim twoarry(1 To 5)
Dim flag As Boolean, flag1 As Boolean
flag1 = False
flag = False
hbpos = 0
lastpos = 0
count1 = 0

bigcount = 0
If maxcount = 2 And ctrlweednum = 2 Then
    lstresult.AddItem herblist(finalhbposition) + " can control "
    For j = 1 To wnum
        If ctrlweed(j) = 1 Then
            weedname = weed(j)
            lstresult.AddItem weedname
        End If
    Next j
End If

If maxcount = 2 And ctrlweednum = 3 Then
    lstresult.AddItem herblist(finalhbposition) + " can control"
    For j = 1 To wnum
        If matrix(finalhbposition, j) = 1 Then
            lstresult.AddItem weed(j)
        End If
    Next j

    lstresult.AddItem ""

    For i = 1 To q
```

```

For j = 1 To wnum
    If flag1 = False And matrix(i, j) = 1 And ctrlweed(j) = 1 _
        And matrix(finalhbposition, j) <> 1 Then
        lstresult.AddItem herblist(i) + " can control "
        lstresult.AddItem weed(j)
        flag1 = True
    End If
Next j
Next i

```

End If

```

If maxcount = 2 And ctrlweednum = 4 Then
    lstresult.AddItem herblist(finalhbposition) + " can control"
    For j = 1 To wnum
        If matrix(finalhbposition, j) = 1 Then
            lstresult.AddItem weed(j)
        Else: twoarr(j) = 1
        End If
    Next j

```

```

    lstresult.AddItem ""

```

```

ReDim herbpos(1 To q) As Integer

```

```

For i = 1 To q

```

```

    For j = 1 To wnum

```

```

        If matrix(i, j) = 1 And ctrlweed(j) = 1 And herblist(i) <> herblist(finalhbposition) _
            And twoarr(j) = 1 Then

```

```

            count1 = count1 + 1

```

```

            hbpos = i

```

```

        End If

```

```

    Next j

```

```

    If count1 > bigcount Then

```

```

        bigcount = count1

```

```

        lastpos = hbpos

```

```

    End If

```

```

    count1 = 0

```

```

Next i

```

```

'lstresult.AddItem bigcount

```

```

If bigcount = 2 And ctrlweednum = 4 Then

```

```

    lstresult.AddItem herblist(lastpos) + " can control "

```

```

For i = 1 To wnum
    For j = 1 To q
        If matrix(j, i) = 1 And ctrlweed(i) = 1 And herblist(j) = herblist(lastpos) Then
            lstresult.AddItem weed(i)
        End If
    Next j

Next i
End If

If bigcount = 1 And ctrlweednum = 4 Then
    For i = 1 To q
        For j = 1 To wnum
            If matrix(i, j) = 1 And ctrlweed(j) = 1 And herblist(i) <> herblist(finalhbposition) _
                And twoarr(j) = 1 Then
                lstresult.AddItem herblist(i) + " can control "
                lstresult.AddItem weed(j)
            End If
        Next j
    Next i
End If
End If

If maxcount = 2 And ctrlweednum = 5 Then
    lstresult.AddItem herblist(finalhbposition) + " can control"
    For j = 1 To wnum
        If matrix(finalhbposition, j) = 1 Then
            lstresult.AddItem weed(j)
        Else: twoarry(j) = 1
        End If
    Next j

lstresult.AddItem ""

For i = 1 To q
    For j = 1 To wnum

        If matrix(i, j) = 1 And ctrlweed(j) = 1 And herblist(i) <> herblist(finalhbposition) _
            And twoarry(j) = 1 Then

            count1 = count1 + 1
            hbpos = i
        End If
    Next j
    If count1 > bigcount Then

```

```

        bigcount = count1
        lastpos = hbpos
    End If
    count1 = 0
Next i

lstresult.AddItem bigcount
If bigcount = 2 And ctrlweednum = 5 Then
    lstresult.AddItem herblist(lastpos) + " can control "
    For i = 1 To wnum
        For j = 1 To q
            If matrix(j, i) = 1 And ctrlweed(i) = 1 And herblist(j) = herblist(lastpos) Then
                lstresult.AddItem weed(i)
            Else: twoarray(i) = 1
            End If
        Next j
    Next i

lstresult.AddItem ""

' print out the fifth herbicide and the controlled weed
For i = 1 To q
    For j = 1 To wnum
        If flag1 = False And matrix(i, j) = 1 And ctrlweed(j) = 1 _
            And matrix(finalhbposition, j) <> 1 And matrix(lastpos, j) <> 1 Then
            lstresult.AddItem herblist(i) + " can control "
            lstresult.AddItem weed(j)
            flag1 = True
        End If
    Next j
Next i
End If

If bigcount = 1 And ctrlweednum = 5 Then
    For i = 1 To q
        For j = 1 To wnum
            If matrix(i, j) = 1 And ctrlweed(j) = 1 And herblist(i) <> herblist(finalhbposition) _
                And twoarray(j) = 1 Then
                lstresult.AddItem herblist(i) + " can control "
                lstresult.AddItem weed(j)
            End If
        Next j
    Next i
End If

```



```
End If
End Sub
```

```
Public Sub threeweeds(wnum As Integer, q As Integer, matrix() As Integer, herblist() As
String, _
weed() As String, maxcount As Integer, finalhbposition As Integer, ctrlweed() As
Integer, _
nctrlweed() As Integer, ctrlweednum As Integer, nctrlweednum As Integer)
```

```
' The maximum number of the weeds controlled by individual herbicide is three
Dim weedname As String
Dim num As Integer, hbpos As Integer, count1 As Integer, bigcount As Integer, lastpos
As Integer
Dim twoarr(1 To 5) As Integer
Dim twoarry(1 To 5)
Dim flag As Boolean, flag1 As Boolean
flag1 = False
flag = False
hbpos = 0
lastpos = 0
count1 = 0
bigcount = 0
lstresult.AddItem ""
lstresult.AddItem herblist(finalhbposition) + " can control three weed"
If maxcount = 3 And ctrlweednum = 3 Then
lstresult.AddItem herblist(finalhbposition) + " can control "
For j = 1 To wnum
    If ctrlweed(j) = 1 Then
        weedname = weed(j)
        lstresult.AddItem weedname
    End If
Next j
End If
```

```
If maxcount = 3 And ctrlweednum = 4 Then
    lstresult.AddItem herblist(finalhbposition) + " can control"
    For j = 1 To wnum
        If matrix(finalhbposition, j) = 1 Then
            lstresult.AddItem weed(j)
        End If
    Next j
```

```
lstresult.AddItem ""
```

```
For i = 1 To q
```

```

For j = 1 To wnum
    If flag1 = False And matrix(i, j) = 1 And ctrlweed(j) = 1 _
        And matrix(finalhbposition, j) <> 1 Then
        lstresult.AddItem herblist(i) + " can control "
        lstresult.AddItem weed(j)
        flag1 = True
    End If
Next j
Next i
End If

If maxcount = 3 And ctrlweednum = 5 Then
    lstresult.AddItem herblist(finalhbposition) + " can control"
    For j = 1 To wnum
        If matrix(finalhbposition, j) = 1 Then
            lstresult.AddItem weed(j)
        Else: twoarr(j) = 1
        End If
    Next j

    lstresult.AddItem ""

    ReDim herbpos(1 To q) As Integer
    For i = 1 To q
        For j = 1 To wnum

            If matrix(i, j) = 1 And ctrlweed(j) = 1 And herblist(i) <> herblist(finalhbposition) _
                And twoarr(j) = 1 Then

                count1 = count1 + 1
                hbpos = i
            End If
        Next j
        If count1 > bigcount Then
            bigcount = count1
            lastpos = hbpos
        End If
        count1 = 0
    Next i

    lstresult.AddItem bigcount
    If bigcount = 2 And ctrlweednum = 5 Then
        lstresult.AddItem herblist(lastpos) + " can control "
        For i = 1 To wnum
            For j = 1 To q

```

```

        If matrix(j, i) = 1 And ctrlweed(i) = 1 And herblist(j) = herblist(lastpos) _
        And twoarr(i) = 1 Then
            lstresult.AddItem weed(i)
        End If
    Next j

Next i
End If

If bigcount = 1 And ctrlweednum = 5 Then
    For i = 1 To q
        For j = 1 To wnum
            If matrix(i, j) = 1 And ctrlweed(j) = 1 And herblist(i) <> herblist(finalhbposition) _
            And twoarr(j) = 1 Then
                lstresult.AddItem herblist(i) + " can control "
                lstresult.AddItem weed(j)
            End If
        Next j
    Next i
End If
End If

End Sub

Public Sub fourweed(wnum As Integer, q As Integer, matrix() As Integer, herblist() As
String, _
weed() As String, maxcount As Integer, _
finalhbposition As Integer, ctrlweed() As Integer, nctrlweed() As Integer, _
ctrlweednum As Integer, nctrlweednum As Integer)
'four weeds can be controlled by one herbicide
Dim weedname As String
num = ctrlweednum - maxcount
If maxcount = 4 And ctrlweednum = 4 Then
    lstresult.AddItem herblist(finalhbposition) + " can control "
    For j = 1 To wnum
        If ctrlweed(j) = 1 Then
            weedname = weed(j)
            lstresult.AddItem weedname
        End If
    Next j
End If

lstresult.AddItem ""

```

```

If maxcount = 4 And ctrlweednum = 5 Then
    lstresult.AddItem herblst(finalhbposition) + " can control"
    For j = 1 To wnum
        If matrix(finalhbposition, j) = 1 Then
            lstresult.AddItem weed(j)
        End If
    Next j

lstresult.AddItem ""

For i = 1 To q
    For j = 1 To wnum
        If flag1 = False And matrix(i, j) = 1 And ctrlweed(j) = 1 _
            And matrix(finalhbposition, j) <> 1 Then
            lstresult.AddItem herblst(i) + " can control "
            lstresult.AddItem weed(j)
            flag1 = True
        End If
    Next j
Next i
End If

End Sub

Public Sub fiveweed(wnum As Integer, q As Integer, matrix() As Integer, herblst() As
String, _
weed() As String, maxcount As Integer, _
finalhbposition As Integer, ctrlweed() As Integer, nctrlweed() As Integer, _
ctrlweednum As Integer, nctrlweednum As Integer)
' five weeds can be controlled by one herbicide
Dim weedname As String
lstresult.AddItem herblst(finalhbposition) + " can control five weed"
If maxcount = 5 And ctrlweednum = 5 Then
    lstresult.AddItem herblst(finalhbposition) + " can control "
    For j = 1 To wnum
        If ctrlweed(j) = 1 Then
            weedname = weed(j)
            lstresult.AddItem weedname
        End If
    Next j
End If

End Sub

Private Sub cmdReturn_Click()
frmSelectHerb.Hide

```

```
frminterface.Show
End Sub
```

```
Public Sub seekmaxcount(wnum As Integer, q As Integer, matrix() As Integer, herblist()
As String, _
weed() As String, maxcount As Integer, _
finalhbposition As Integer, ctrlweed() As Integer, nctrlweed() As Integer, _
ctrlweednum As Integer, nctrlweednum As Integer, hbposition As Integer, _
count As Integer, str1 As String, str2 As String, str3 As String)
```

```
'This sub procedure is to find out the maximum number of the weeds controlled
'by individual herbicide
```

```
Dim i As Integer, j As Integer
Dim MaxNum As String
```

```
For i = 1 To wnum
    For j = 1 To q
        If matrix(j, i) = 1 Then
            ctrlweed(i) = 1
        End If
    Next j
    If ctrlweed(i) = 1 Then
        ctrlweednum = ctrlweednum + 1
    End If
    If ctrlweed(i) <> 1 Then
        nctrlweed(i) = 1
        nctrlweednum = nctrlweednum + 1
    End If
Next i
```

```
For i = 1 To q
    For j = 1 To wnum
        If matrix(i, j) = 1 Then
            count = count + 1
            hbposition = i
        End If
    Next j
    If count > maxcount Then
        maxcount = count
        finalhbposition = hbposition
    End If
    count = 0
Next i
```

```
If nctrlweednum <> 0 And maxcount <> 0 Then
    lstresult.AddItem ""
```

```

    lstresult.AddItem str1
    lstresult.AddItem str3
    For i = 1 To wnum
        If nctrlweed(i) = 1 Then
            lstresult.AddItem weed(i)
        End If
    Next i
End If

If ctrlweednum <> 0 Then
    lstresult.AddItem ""
    lstresult.AddItem str2
    lstresult.AddItem str3
    For j = 1 To wnum
        If ctrlweed(j) = 1 Then
            lstresult.AddItem weed(j)
        End If
    Next j
End If

If maxcount = 0 Then
    MaxNum = "zero"
End If
If maxcount = 1 Then
    MaxNum = "one"
End If
If maxcount = 2 Then
    MaxNum = "two"
End If
If maxcount = 3 Then
    MaxNum = "three"
End If
If maxcount = 4 Then
    MaxNum = "four"
End If
If maxcount = 5 Then
    MaxNum = "five"
End If
    lstresult.AddItem ""
    lstresult.AddItem "The maximum number of weeds controlled "
    lstresult.AddItem "by individual herbicide is " + MaxNum
    lstresult.AddItem ""
End Sub

```

Literature Cited

- Adamski, J. J., C.Hommel and K. T. Finnegan. Microsoft Access 97. Course Technology. Cambridge, MA 02142.
- Aho, A.V., C. Beeri and J. D. Ullman. The theory of joins in relational databases. ACM Transactions on Database Systems, Vol. 4, No. 3, September 1979, 297-314.
- Bernstein, P.A. Synthesizing Third Normal Form Relations from Functional Dependencies. ACM Transactions on Database Systems, Vol. 1, No. 4, December 1976, 277-298.
- Chen, P.P. The Entity-Relationship Model – Toward a Unified View of Data. ACM Transactions on Database Systems, Vol. 1, No. 1, March 1976, 9-36.
- Caldeira, C. P. and P. A. Pinto. Linking DSSAT V3 to a relational database: the AGROSYS-DSSAT interface. Computers and Electronics in Agriculture. 21: 69-77.
- Chakravarthy, U. S., J. Grant and J. Minker. Logical-Based Approach to Semantic Query Optimization. ACM Transactions on Database Systems, Vol. 15, No. 2, June 1990, 162-207.
- Codd, E. F. A Relational Model for Large Shared Data Bank. Communications of the ACM, Vol. 13, No. 6, June 1970, 377-387.
- Codd, E. F. Extending the Database Relational Model to Capture More Meaning. ACM Transactions on Database Systems, Vol. 4, No. 4, December 1979, 397-434.
- Gould, F. 1995. Comparison between resistance management strategies for insect and weeds. Weed Technol. 9:830-839.

Hall, R.M., C. J. Swanton and G. W. Anderson. 1992. The critical period of weed control in grain corn (*Zea mays*). *Weed Sci.* 40:441-447.

Humbyrg, N. E., et al. 1989. *Herbicide Handbook of the Weed Science Society of America*. Sixth Edition, Weed Science Society of America, 309 West Clark Street, Champaign, Illinois 61820.

Li, S. and Z. Gao. 1990. *The Eco-economical principles of weed management in crop field*. Beijing Agricultural University Press 1-163.

Maxwell, B. D., M. L. Roush, and S. R. Radosevich. 1990. Predicting the evolution and dynamics of herbicide resistance in weed populations. *Weed Technol.* 3:2-13.

Muelier-Warrant, G. W. 1991. Format of a weed control database. *Weed Technol.* 5:221-228.

Pan, X., Hesketh, J. D. and M. G. Huck. 1998. A web interface to databases associated with a plant growth simulator. *Computers and Electronics in Agriculture*. 21: 207-217.

Peter, R. and C. Coronel. 1997. Third Edition. *Database Systems Design, Implementation, and Management*. Course Technology. 1 Main Street, Cambridge, MA 02142.

Peterson, D. E. and D. L. Regehr. 1995. Postemergence broadleaf weed control in soybean. *NCWSS Res. Report*. 51: 279.

Radosevich, S. R. 1987. Methods to study interactions among crops and weeds. *Weed Technol.* 1:190-198.

Ramakrishnan, R. 1998. *Database Management Systems*. WCB/McGraw-Hill.

- Regehr, D. L., et al. 1995. Chemical Weed Control. Agricultural Experiment Station, Kansas State University, Manhattan 66506.
- Schneider, D. I. 1998. An Introduction to Programming Using Visual Basic 5.0. Third Edition. Prentice Hall. Upper Saddle River, NJ 07458.
- Schneider, D. I. 1998. An Introduction to Programming Using Visual Basic 6.0. Fourth Edition. Prentice Hall. Upper Saddle River, NJ 07458.
- Shaffer, M. J., M. K. Brodahl. 1998. Rule-based management for simulation in agricultural decision support system. Computers and Electronics in Agriculture. 21:135-152.
- Silberschatz, A., H. F. Korth and S. Sudarshan. 1998. Third Edition. Database System Concepts. WCB/McGraw Hill.
- Stigliani, L, G. Santospirito, N. Cardinale and C. Resina. 1996. A Relational database as decision support system in chemical weed control. Weed Technol. 10:781-794.
- Ullman, J. D. 1988. Principles of Database and Knowledge – Base Systems, Volume I. Computer Science Press, 1803 Research Boulevard, Rockville, MD 20850.
- Ullman, J. D. 1989. Principles of Database and Knowledge – Base Systems, Volume II. Computer Science Press, 1803 Research Boulevard, Rockville, MD 20850.
- Van Acker, R. C., C. J. Swanton and S. F. Weise. 1993. The critical period of weed control in soybean [*Glycine max* (L.) Merr.]. Weed Sci. 41:194-200.
- Van Evert, F. K., E. J. A. Spaans, S. D. Krieger, J. V. Carlis and J. M. Baker. 1991. A database for agroecological research data: I. Data model. Agron. Jour. 91:54-62.

Van Evert, F. K., E. J. A. Spaans, S. D. Krieger, J. V. Carlis and J. M. Baker. 1991. A database for agroecological research data: II. A relational implementation. *Agron. Jour.* 91:62-71.

Wilkerson, G. G., S.A.Modena, and H.D.Coble. 1991. Herb: Decision model for Postemergence Weed control in soybean. *Agron. Jour.* 83: 417-423.

Zhang, L. and K. Liu. Towards a Relational Model for Exclusively Disjunctive Information. 21st Annual Computer Conference February 16-18, 1993. Indiana Convention Center, Indianapolis, Indiana proceedings. 143-150.

2

VITA

Zhuping Gao

Candidate for the Degree of
Master of Science

Thesis: RELATIONAL DATABASE AND OPTIMAL HERBICIDE SELECTION IN
CHEMICAL WEED CONTROL

Major Field: Computer Science

Biographical:

Education: Received Bachelor of Science degree in Agronomy from Shanxi Agricultural University, Shanxi, China in January 1982. Received Master of Science degree in Agronomy from Beijing Agricultural University, Beijing, China in July 1987. Received Master of Science degree in Agronomy from Kansas State University, Manhattan, Kansas in May 1997. Completed the requirements for the Master of Science degree with a major in Computer Science at Oklahoma State University in July, 2000.

Experience: Employed by Oklahoma State University, Department of Computer Science as a teaching assistant, 08/1999 to present; employed by Oklahoma State University, Department of Computing and Information Services as a computer assistant, 04/1999 to 01/2000; employed by Kansas State University, Department of Agronomy as a graduate research assistant, 6/1995 to 05/1997; do research work as a visiting scientist at Kansas State University, Department of Agronomy, 12/1993 to 05/1995; employed by Shanxi Academy of Agricultural Science of China, Soil and Fertilizer Institute as an assistant professor, 08/1987 to 11/1993; employed by Beijing Agricultural University, Department of Agronomy as a graduate research assistant, 09/1984 to 07/1987; Employed by Shanxi Agricultural University, Department of Agronomy as a teaching assistant, 02/1982 to 08/1984.