

CHAPTER 1

INTRODUCTION

1.1 COURSE OBJECTIVE

The student shall be capable of identifying a problem related to the field of Computer Science and carry out a mini project on the problem defined by that project using python programming language and with database concepts. Each student is expected to do the mini project individually to make themselves habituated on using python programming language and get experience in utilizing SQL commands. The students are expected to choose topics wisely to which area they are actually interested in. The project topic must also be interesting enough to show that this area is what our project must be focused on. The project must include GUI with database related with it. The student must have proper knowledge about how they have implemented their code and how does it work. They must have proper knowledge about what they have presented. Since, python and DBMS are the most used concepts on most of the fields, students are expected to have adequate information on them.

1.2 PROBLEM STATEMENT

The problem related to this mini project is to connect any database using python program. The data provided in the database should be accessible and should be manipulate able. And the data provided in the database should not be redundant. We should have as less no of NULL values as possible. The program should make utmost utilization of SQL queries to make database standard. Also, it challenges the concept of security of the system for the admin mode. The project on Agronomy is completely based on the Data Base Management System concept. The project is about storing the data provided by the satellite (user in our program) and utilizing those results to give instruction on suitable crops, pesticides and equipment that

can be helpful for a use. This program manipulates the database related to weather and agriculture. Some of the widgets of tkinter are used for GUI environment.

Here, the program provides two options: USER & ADMIN. Admins are allowed to manipulate the data in the database to provide consistent data to the user. While the users are provided a user-friendly environment to communicate with the database. The users are asked to enter their location and its current humidity, temperature and soil p^h value. Then the program suggests the crops that are suitable for that location with adequate information on it.

1.3 OUTCOME OF THE PROJECT WORK CARRIED

- The admin is able to manipulate the database.
- The user is provided with adequate knowledge about what crops can be cultivated.
- A brief information on Pesticides and equipment is also given.

CHAPTER 2

REQUIREMENTS AND DESIGN

3.1 Hardware requirements

- Processor: Intel Pentium 4 or up
- RAM: 1 GB or more
- Hard disk: 40 GB or more

3.2 Software requirements

- Jupyter Notebook
- Mysql
- Google chrome or any web browser
- Db Browser (SQLite)
- Windows 7 or later versions

CHAPTER 3

Data Models and ER Model

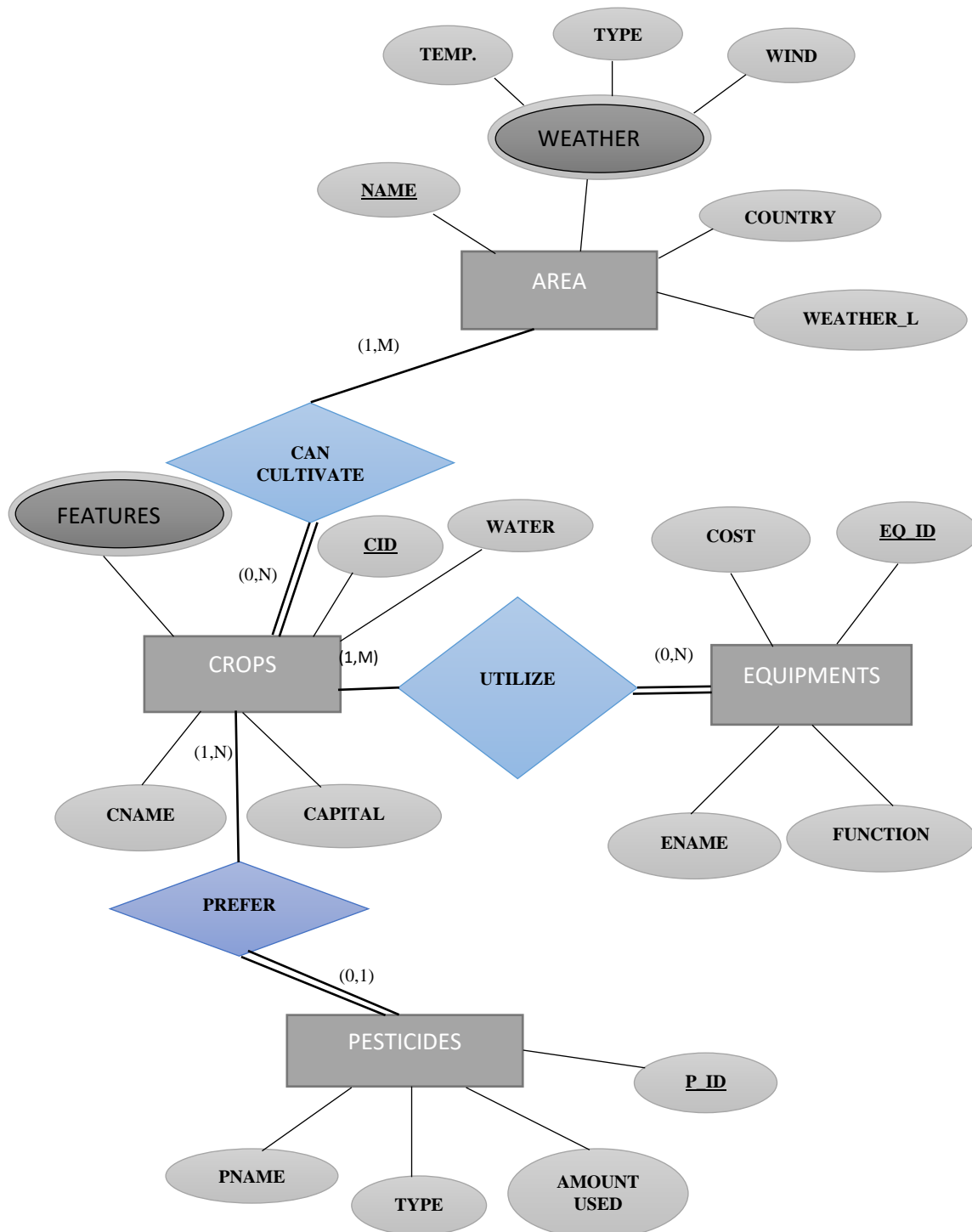


Fig. 3.1: ER diagram

3.1 Entity and Attributes

3.1.1 Entity

An entity is a real-world object that are represented in a database. It is one of the main components of ER Diagram Data. They are represented by a rectangular box in ER Diagram. A separate table is made for each of these entities while creating database.



Fig. 3.2: Entity

3.1.2 ATTRIBUTES

An attribute is a characteristic/property of any entity present in a database. In ER Diagram, they are represented by oval. There are different types of attributes and are represented by ovals. Only for multivalued attribute a separate table is created.

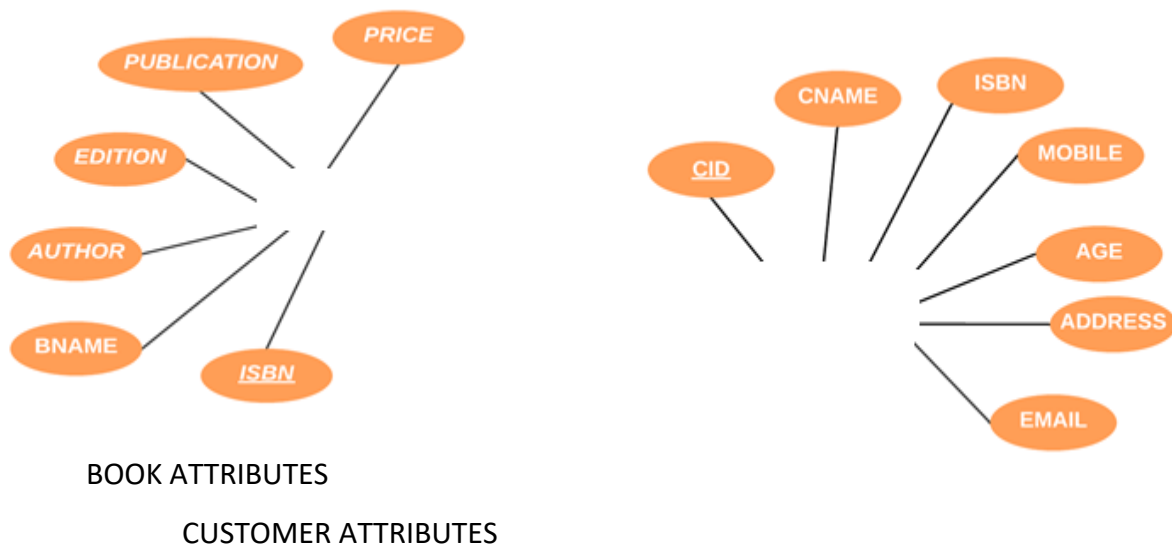


Fig. 3.3: attributes

CHAPTER 4

Relational Model Concepts and Schema

A relational model represents the database with a collection of tables/relations. In DBMS, a table is referred as a relation. Any table in a database holds all the data that are related to an entity. The data stored in a table denotes a real world-entity or relationship.

Relational model contains following main points:

- **Attributes:** Attributes are the properties/characteristics of an entity. Attributes are each column or field in a relation.
- **Tables:** In relational model, a relation is saved in a tabular format. Columns in the table are the entities and rows are the records.
- **Relation Schema:** A relation schema describes each of the relation along with its attributes.
- **Tuples:** Tuples represents the rows in a relation that is each record in a table.
- **Degree:** The degree of a relation is the total number of attributes a relation consists of.
- **Cardinality:** Cardinality refers to the total number of rows present in the table.
- **Column:** The column stores specific values for an attribute.
- **Relation instance:** Relation instance consists of a finite set of tuples in a relational database management system. They do not have any duplicate tuples.
- **Relation key:** Every row has one, two or many key attributes which are called relation key.
- **Attribute domain:** Every attribute in a relation have some predefined values and scope which is known as attribute domain.

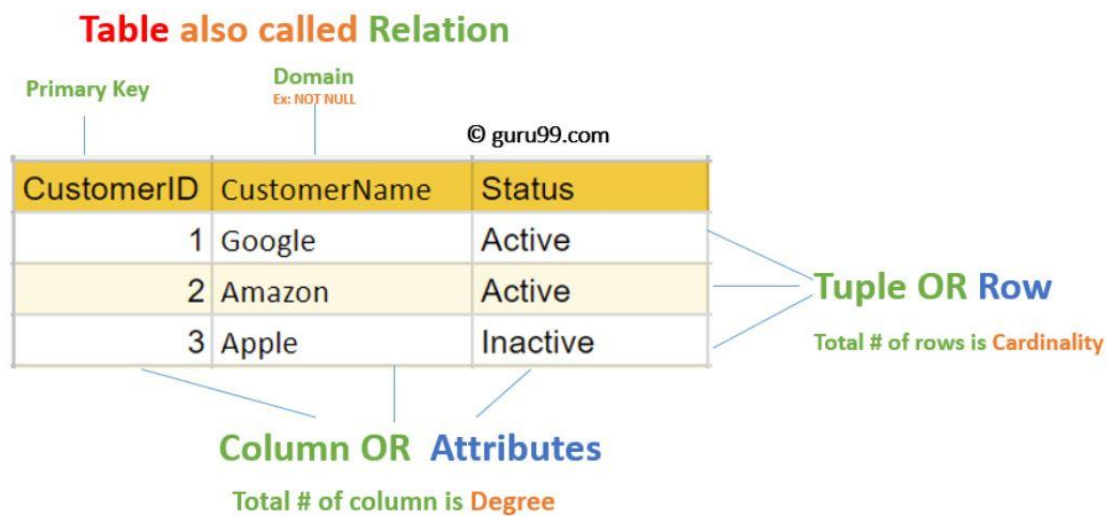


Fig. 4.1: Components of RDBMS

4.1 Relational integrity constraints:

For a valid relation some conditions should be met, which are satisfied using relational integrity constraints. In Relational database management system, integrity constraints are divided into three parts:

- Domain constraints
- Key constraints
- Referential integrity constraints

4.1.1 Domain constraints:

If an attribute value is not appearing in the corresponding domain or it is of inappropriate data type then the domain constraint is violated. Domain constraint is used to specify that within each attribute the value must be unique.

Example:

Create domain StudentName CHECK (value not NULL);

The given example shows creation of domain constraint such that StudentName is not NULL.

4.1.2 Key constraints:

Key is an attribute which can uniquely identify a tuple. The value of key/attribute should be unique for each record/tuple.

Example: In a Student database, StudentID is the key of Student table since StudentID is unique for all student.

4.1.3 Referential integrity constraints:

Referential integrity constraint defines the concept of Foreign key. It is used when an attribute that is primary key of one relation has to be added to another relation.

Example:

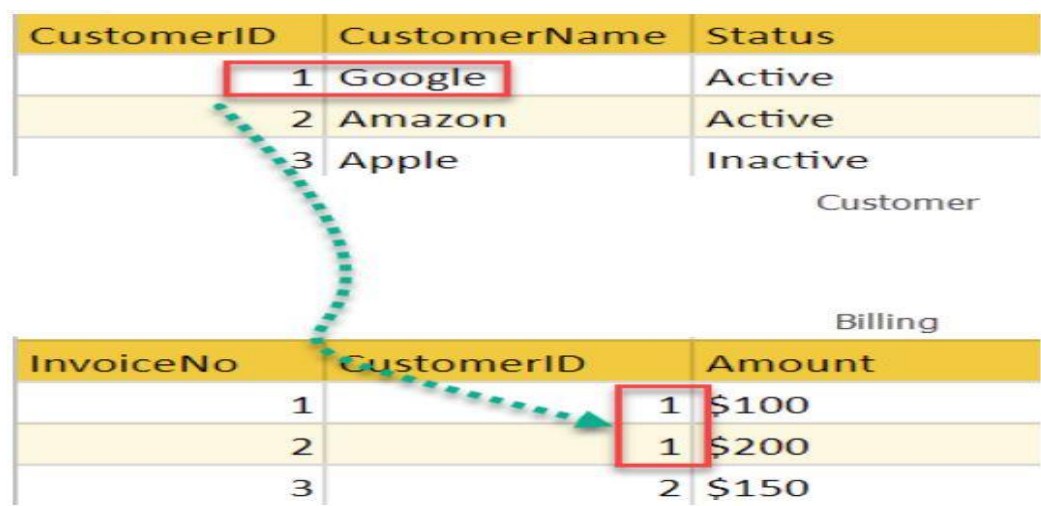


Fig 4.2: Primary key and Foreign key

In the above figure, CustomerID is brought to Billing table as a foreign key which is the Primary key of Customer table.

4.2 Operations in Relational Model:

There are four basic operations that can be performed on RDBMS:

- **Insertion:** The insertion operation gives values of attributes for a new record which is to be inserted into the table.
- **Update:** The update operation is used to update the values selected records.
- **Delete:** A condition on the query selects the tuples that are to be deleted.
- **Select:** Select operation is used to view the data present in table by filtering out required values.

4.3 Advantages of Relation Model:

- **Simplicity:** A Relational Model presents simpler model rather than hierarchical and network.
- **Structural Independency:** Since relational database is only concerned with data and structure, the performance of the model can be improved.
- **Easy to Use:** Tables with rows and columns are easy to understand, so it is also easy to use.
- **Query Capability:** It helps high level language like SQL to avoid database complexity.
- **Data Independence:** The structure of a database can be modified easily without having to change any application.
- **Scalable:** To enhance the usability of a database, the number of records and the files should be increased.

4.4 RELATIONAL SCHEMA

A relational database program is a standard tool for storing and analysing data with the proper management. Examples Microsoft SQL Server, PostgreSQL, MySQL and various products from companies such as Oracle and IBM. A relational schema outlines the concept of database relationships and structure in a relational database program. It can be displayed

graphically or written in the Structured Query Language (SQL) used to build tables in a relational database.

WEATHER:

ZIPCODE	WEATHER TYPE	TEMPERATURE	WIND
---------	--------------	-------------	------

AREA:

ZIPCODE	CITY	COUNTRY	WEATHER LINK
---------	------	---------	--------------

PESTICIDE:

PESTICIDE ID	PESTICIDE NAME	TYPE	AMOUNT TO USE
--------------	----------------	------	---------------

CROP:

CROP ID	CROP NAME	CAPITAL	SUITABLE TEMPERATURE	SOIL PH	PESTICIDE ID
---------	-----------	---------	----------------------	---------	--------------

USES:

CR ID	USE
-------	-----

EQUIPMENT:

EQUIPMENT ID	EQUIP NAME	COST	FUNCTION
--------------	------------	------	----------

UTILIZE:

CROP ID	EQ ID
---------	-------

REQUIREMENT:

LOW TEMPERATURE	HIGH TEMPERATURE	SOIL PH	CROP ID
-----------------	------------------	---------	---------

CHAPTER 5

SQL

SQL stands for Structured Query Language. It is a standard computer language for managing the relational database and manipulation of data. There are various SQL commands which are listed below:

5.1 DDL commands

DDL stands for Data Definition Language. There are various DDL commands which are listed below:

5.1.1 CREATE

This command is used to create a table.

Syntax: CREATE TABLE VALUES (column1, column2,...); e.g.: CREATE TABLE IF NOT EXISTS 'WEBSITE'(WID INT PRIMARY KEY AUTO INCREMENT, LINK VARCHAR(20) NOT NULL, WNAME VARCHAR(20) NOT NULL);

5.1.2 DROP

This command is used to delete table or the objects from the database.

Syntax: DROP TABLE tablename ;

e.g.: DROP TABLE WEBSITE;

5.1.3 ALTER

This command is used to alter the table structure like table definitions, constraints, delete columns, add columns, etc. It has various options like ADD, MODIFY, CHANGE, DELETE etc.

Syntax: ALTER TABLE tablename command ;

5.1.4 TRUNCATE

This command is used to delete the contents of the table including all the spaces.

Syntax: TRUNCATE TABLE ;

5.2 DQL commands

DQL stands for Data Query Language for performing database queries.

5.2.1 SELECT

This command is used to retrieve the data from the database. It is one of the widely used command and is complex.

Syntax: SELECT columnnames FROM tablename WHERE conditions ;

e.g.: SELECT * FROM WEBSITE WHERE WID=1;

5.3 DML commands

DML stands for Data Manipulation Language and is used to manipulate the rows in the table. The rows in the table can be updated, deleted, inserted, etc.

5.3.1 INSERT

This command is used to insert the tuples into the table.

Syntax: INSERT INTO tablename VALUES(.,...) ;

e.g.: INSERT INTO WEBSITE (LINK,WNAME) VALUES ('www.google.com','google');

5.3.2 UPDATE

This command is used to update the existing values of the tuple. Variable values can be updated using this command.

Syntax: UPDATE tablename SET ="" WHERE condition ;

5.3.3 DELETE

This command is used to delete the tuples from the tables.

Syntax: DELETE FROM tablename WHERE condition;

e.g.: DELETE FROM WEBSITE WHERE WID=10;

5.4 TCL command

TCL stands for Transaction Control Language which deals with the transaction within the database.

5.4.1 COMMIT

This command is used to commit the transaction so that the previous transaction will be successfully saved into the database. Once commit is done it is not possible to rollback.

Syntax: COMMIT;

5.4.2 ROLLBACK

This command is used to rollback undo the transaction if any error occurs.

Syntax: ROLLBACK;

Chapter 6

Python Features

Python is a high-level, general-purpose programming language used vastly in the IT development of apps and programs. In programming, data type is an important concept. Variables can store data of different types, and different types can do different things.

Python has the following built-in data types by default:

1. String
2. List
3. Tuple
4. Set
5. Dictionary

Talking about the feature of python, they are enlisted as below and discussed briefly.

6.1 Easy to code:

Python is high level programming language. Python language is very easy to learn as compared to other language like c, c#, java script, java etc. Also, python is very easy to understand and can do most difficult task easily. Thus, it is preferred mostly by every company and widely used. Anybody can learn python basics in few days. It is also developer-friendly language.

6.2. Free and Open Source:

Python language is freely available at official website. Since, it is open-source, this means that source code is also available to the public. So, it can be downloaded whenever one wants and start coding on it easily.

6.3. Object-Oriented Language:

A programming language that can model the real world is said to be object-oriented. It focuses on objects, and combines data and functions. Contrarily, a procedure-oriented

language revolves around functions, which are code that can be reused. Python supports both procedure-oriented and object-oriented programming which is one of the key python features. It also supports multiple inheritance, unlike Java. A class is a blueprint for such an object. It is an abstract data type, and holds no values. Python supports object oriented language and concepts of classes, objects encapsulation etc.

6.4.GUI Programming Support:

Graphical Users interfaces can be made using a module such as PyQt5, PyQt4, wxPython or Tk in python. PyQt5 is the most popular option for creating graphical apps with Python.

6.5. High Level Language:

Python is a high-level language. When we write programs in python, we do not need to remember the system architecture nor do we need to manage the memory.

6.6. Extensible feature:

Python is an **Extensible** language. We can write our some python code into C or C++ language and also we can compile that code in C/C++ language.

6.7. Python is Portable language:

Python language is also a portable language. For example, if we have python code for windows and if we want to run this code on other platform such as Linux, Unix and Mac then we do not need to change it, we can run this code on any platform.

6.8. Python is Integrated language:

Python is also an Integrated language because we can easily integrated python with other language like C, C++ etc.

6.9. Interpreted Language:

Python is an Interpreted Language. because python code is executed line by line at a time. like other language C, C++, java etc there is no need to compile python code this makes it easier to debug our code. The source code of python is converted into an immediate form called **bytecode**.

6.10. Large Standard Library

Python has a large standard library which provides rich set of module and functions so you do not have to write your own code for every single thing. There are many libraries present in python for such as regular expressions, unit-testing, web browsers etc.

6.11. Dynamically Typed Language:

Python is dynamically-typed language. That means the type (for example- int, double, long etc) for a variable is decided at run time not in advance because of this feature we don't need to specify the type of variable.

Chapter 7

Tkinter Widgets

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy-task.

create a tkinter:

1. Importing the module – tkinter
2. Create the main window (container)
3. Add any number of widgets to the main window
4. Apply the event Trigger on the widgets.

Importing tkinter is same as importing any other module in the python code. Note that the name of the module in Python 2.x is 'Tkinter' and in Python 3.x is 'tkinter'.

Some of the GUI Widgets provided by Tkinter are as follows:

1. Button
2. Label
3. Entry
4. Menu
5. MessageBox

7.1 BUTTON

The **Button** widget is a standard Tkinter widget used to implement various kinds of buttons. Buttons can contain text or images, and you can associate a Python function or method with each button. When the button is pressed, Tkinter automatically calls that function or method.

The button can only display text in a single font, but the text may span more than one line. In addition, one of the characters can be underlined, for example to mark a keyboard shortcut. By default, the **Tab** key can be used to move to a button widget.

The Button widget is used to add buttons in your application. The general syntax is:

“w=Button(master, option=value)” , master is the parameter used to represent the parent window. For Example:

```
b1=Button(root,text="GO ->",bd=7,width = 10,bg='light green',activebackground='WHITE',command=lambda:[prog(),msg()])
b1.place(x=200,y=250)

b2=Button(root,text="Quit",bd=7,width=10,bg='red',fg='WHITE',command=root.destroy)
b2.place(x=340,y=250)
```

Output:



Fig 7.1 Button

When to use the Button Widget:

Simply put, button widgets are used to let the user say “do this now!,” where *this* is either given by the text on the button, or implied by the icon displayed in the button. Buttons are typically used in toolbars, in application windows, and to accept or dismiss data entered into a dialog box.

7.2 LABEL

The Label widget is used to provide a single-line caption for other widgets. It refers to the display box where you can put any text or image which can be updated any time as per the code.

The general syntax is:

“w=Label(master, option=value)” , master is the parameter used to represent the parent window.

Example:

```
fname=Label(root,text="Enter the Folder Path : ",font=('aria',14))  
fname.place(x=110,y=150)
```

Output:

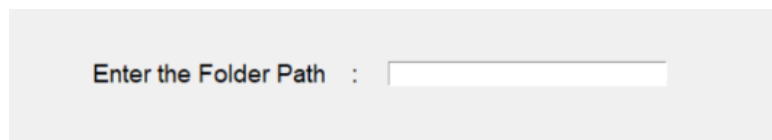


Fig 7.2: Label Widget

7.3 ENTRY

Entry is used to input the single line text entry from the user. For multi-line text input, Text widget is used. The general syntax is:

“w=Entry(master, option=value)” , master is the parameter used to represent the parent window. Example:

```
efname=Entry(root,width=35)  
efname.place(x=340,y=155)
```

Output:

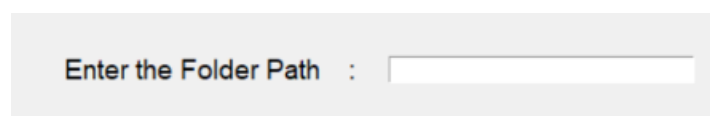


Fig 5.3 Entry Widget

7.4 MENU

The Menu widget is used to provide various commands to a user. These commands are contained inside Menubutton. It is a part of top-down menu which stays on the window all the time. Every menubutton has its own functionality.

The general syntax is:

“w = MenuButton(master, option=value)” , master is the parameter used to represent the parent window.

Menu is of three types: Top Level, Pull Down and Pull Up.

Example:

```
top = Tk()
mb = Menubutton ( top, text = &quot;GfG&quot; )
mb.grid()
mb.menu = Menu ( mb, tearoff = 0 )
mb[&quot;menu&quot;] = mb.menu
cVar = IntVar()
aVar = IntVar()
mb.menu.add_checkbutton ( label = 'Contact', variable = cVar )
mb.menu.add_checkbutton ( label = 'About', variable = aVar )
mb.pack()
top.mainloop()
```

Output:

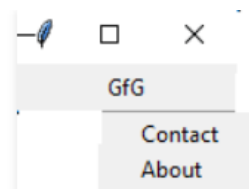


Fig 7.4 Menubutton

7.5 MESSAGE BOX

The Message widget is used to display multiline text fields for accepting values from a user. Its work is same as Label. The general syntax is:

“w = Message(master, option=value)” , master is the parameter used to represent the parent window.

Example:

```
def msg():  
    messagebox.showinfo("Information", "Files have been organized!")
```

Output:

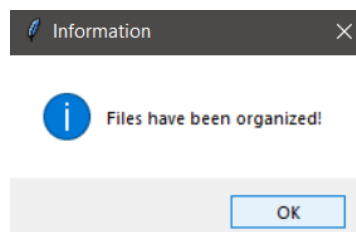


Fig 7.5 MessageBox

7.6 Frame

A frame is rectangular region on the screen which is used to hold widgets, photos, videos, etc. in a window. The frame widget is mainly used as a geometry master for other widgets, or to provide padding between other widgets. Frame widgets are used to group other widgets into complex layouts. They are also used for padding, and as a base class when implementing compound widgets. The frame widget can also be used as a place holder for video overlays and other external processes.

To use a frame widget in this fashion, set the background colour to an empty string (this prevents updates, and leaves the colour map alone), pack it as usual, and use the **window_id** method to get the window handle corresponding to the frame.

```
frame = Frame(width=768, height=576, bg="", colormap="new")  
frame.pack()  
video.attach_window(frame.window_id())
```

CHAPTER 8

IMPLEMENTATION

This program is written in python language. With the help of Tkinter libraries like tkinter, GUI windows are created along with frame, button, labels, etc widgets. SQLITE3 library is imported to connect the database and to use SQL queries.

At first, the user is asked to choose whether they want to login as admin or user. For the admin option, the user is allowed with multiple options like area, weather, pesticides, etc. These options are for the user to insert the values into the corresponding tables. Here, we use web scraping to find the current weather condition of any given location. Ideally, the satellite collects the weather information and stores the data in a database. But here we are allowing an admin to get access of the database. Here, the admins are the only one who are allowed to manipulate the data.

Then the application is deployed and the users are allowed to use the application. The users are provided with user friendly environment to communicate with the database. At first, Area name with their country name asked, then they are asked to enter their soils' pH value. After entering, the program compares the given input with the database value to perform required calculation. And with the current weather and soil fertility, it suggests the suitable crops that can be cultivated in that area. It also suggests pesticides and equipment that can be used. The users are allowed to only retrieve the data from database.

Actually, this program depicts the calculation that happens while prediction of weather. Though all the data may not be accurate, this project was implemented simply to emphasize on the topic of how agriculture can be benefited and uplifted from current condition using current technology.

Chapter 9

OUTPUT SNAPSHOTS

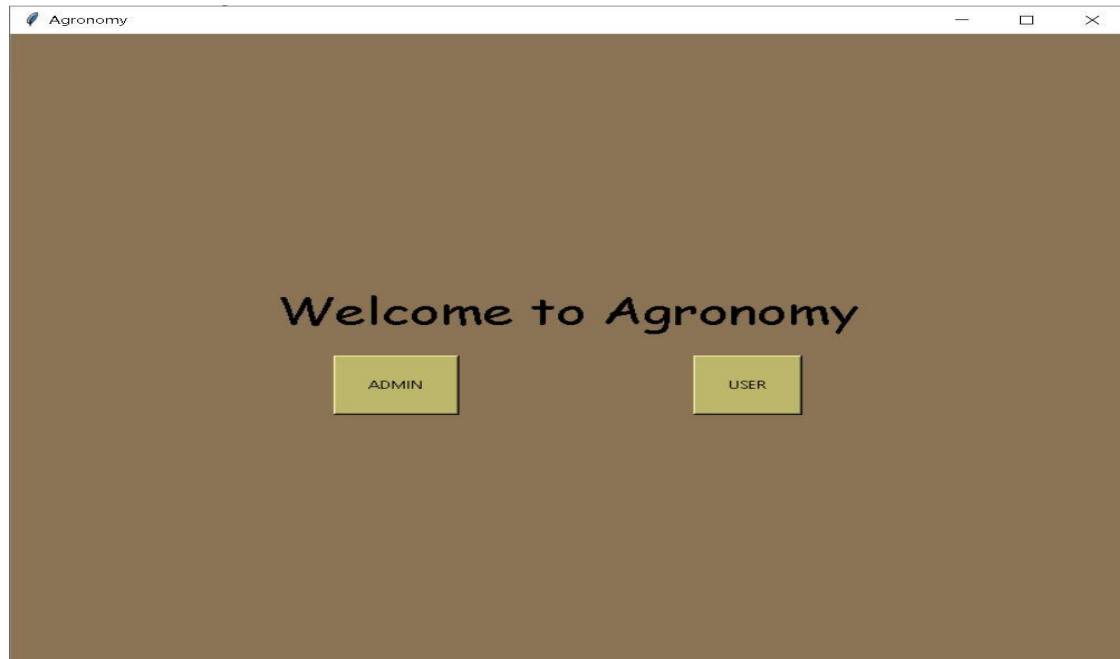


Fig 9.1: Electives

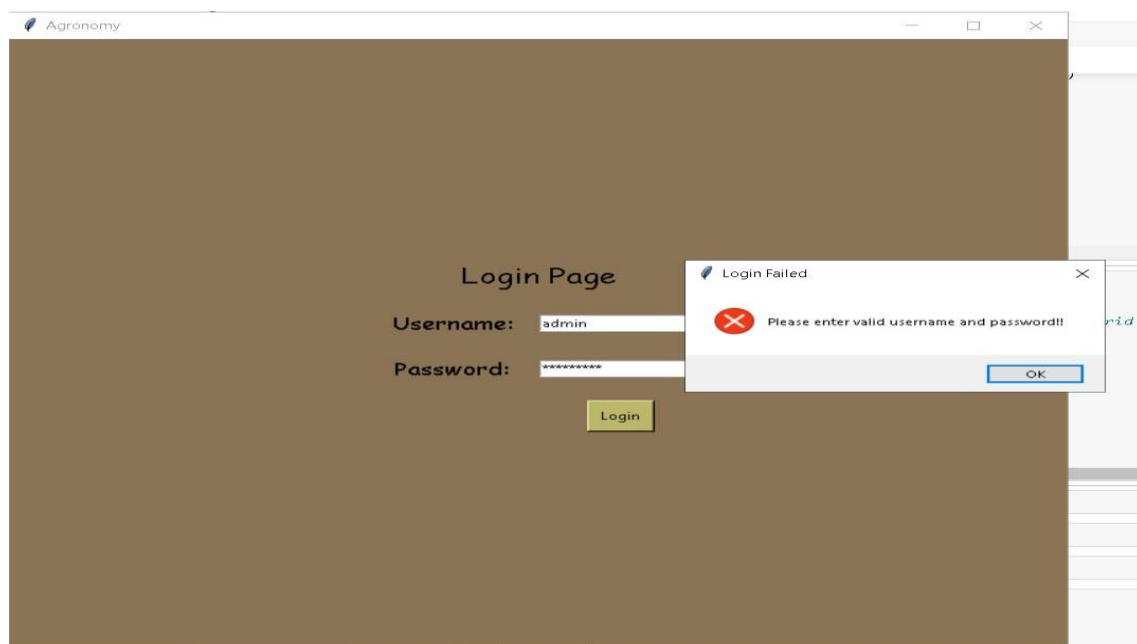


Fig 9.2: Admin Login Page

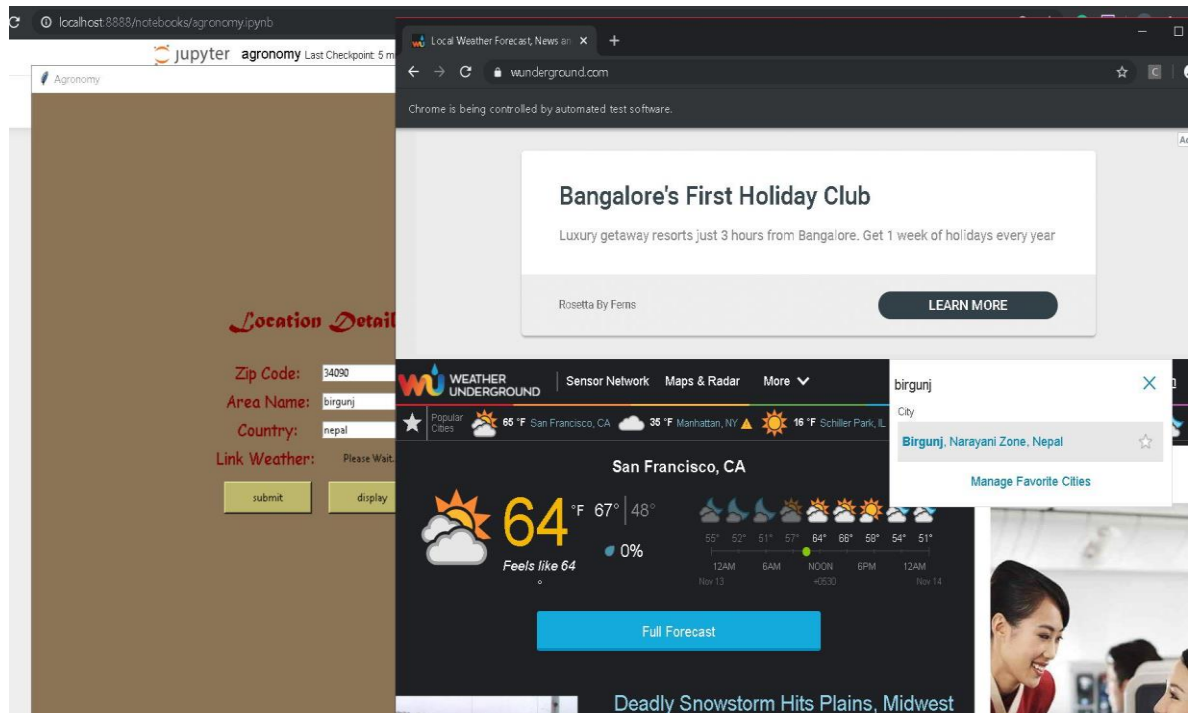


Fig 9.3: Web Scraping

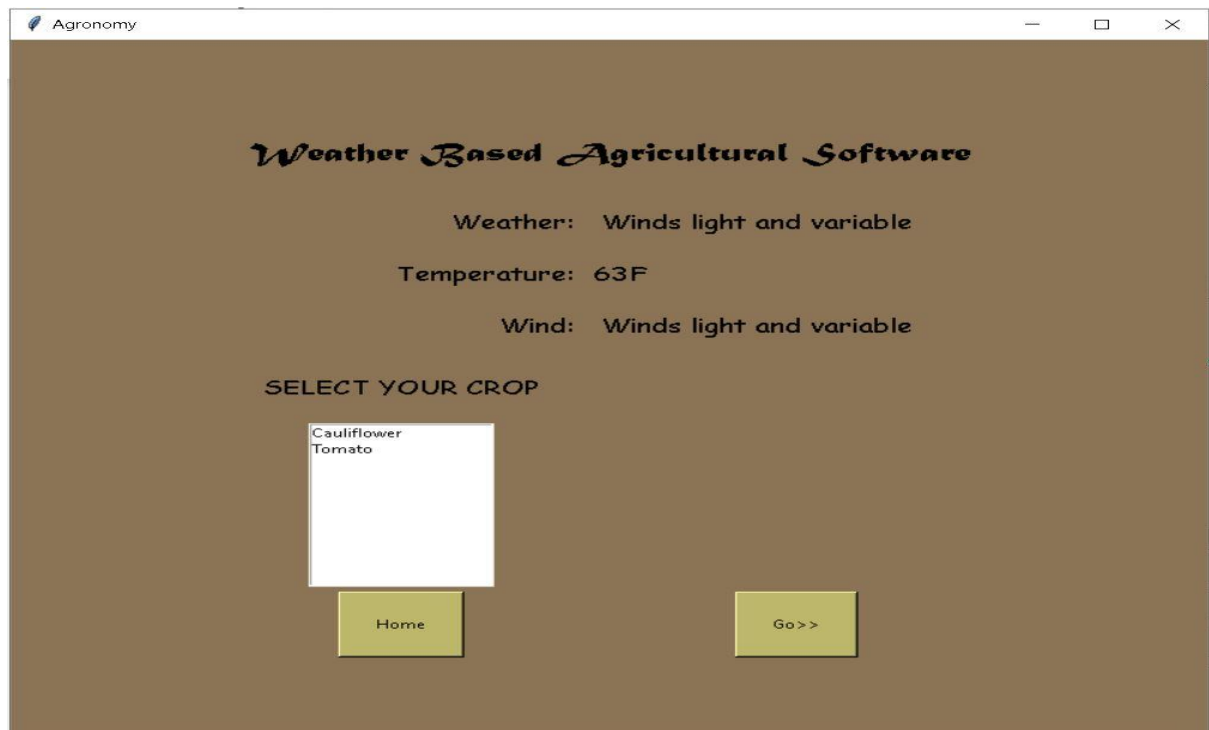


Fig 9.4: User Interface

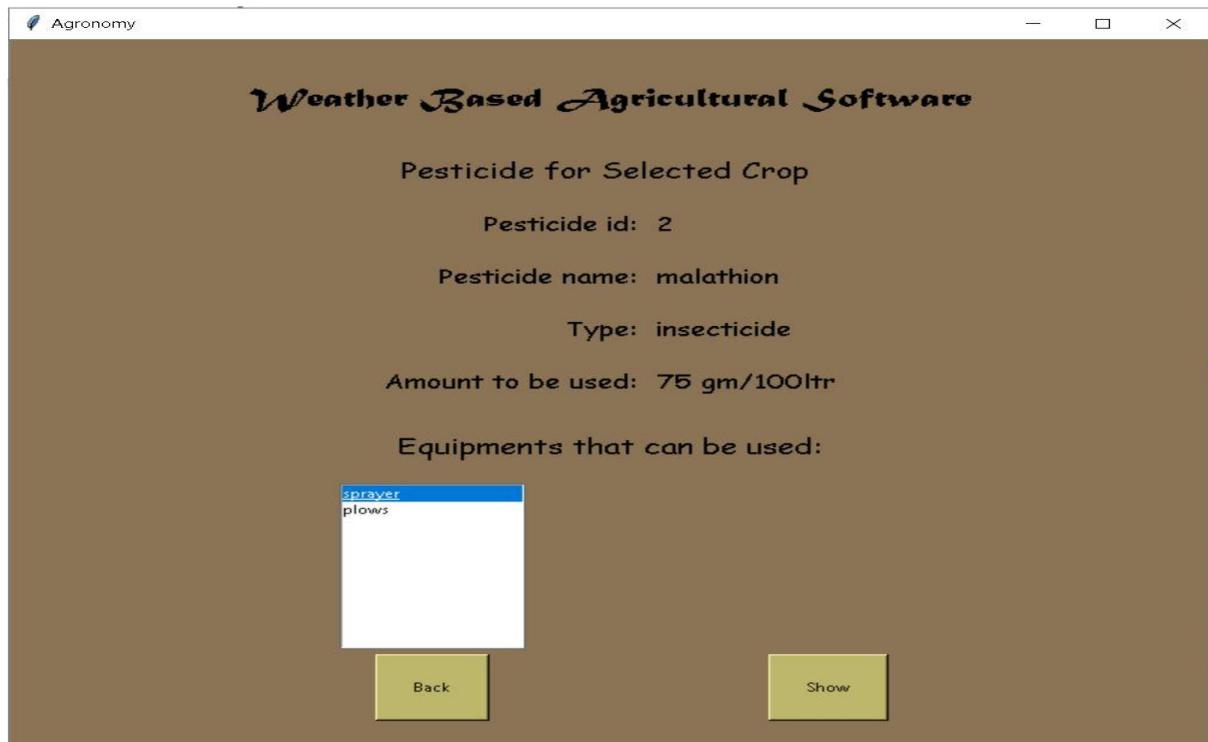


Fig 9.5: Pesticide details

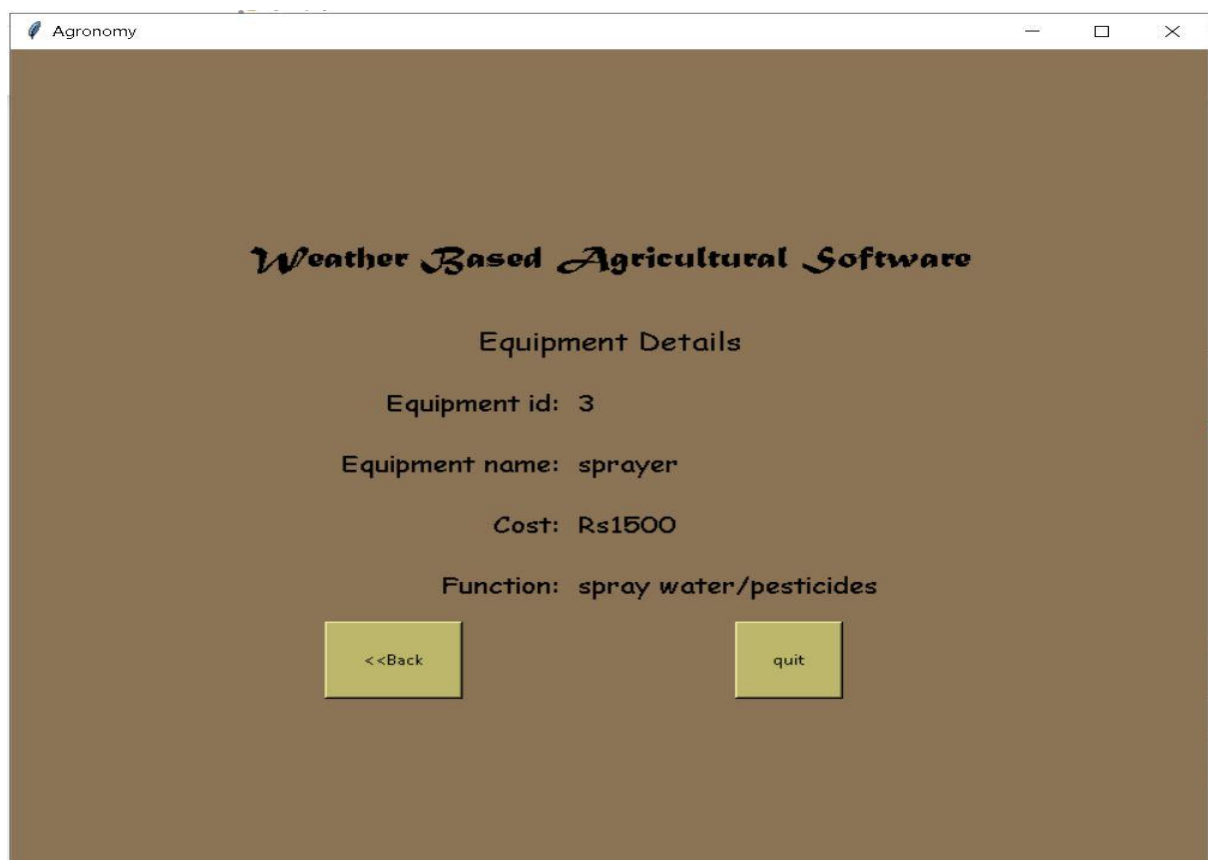


Fig 9.6: Equipment Details

Chapter 10

CONCLUSION

The mini-project on agronomy has helped me to understand the concept of database management system along with its implementation using Python Programming Language which is the main objective of the project. Here, the main aim of this project is to help the user choose what kind of crops are suitable for their land. Though this project was out of topic, it helped us explore more on weather and agriculture field. Though many new programming languages are being developed, python has remained the basic knowledge requirement for most company and recruiters. And since everything are being modernized, the amount of data that needs to be stored has been increasing day by day. So, there is always a need for organizing all those big data in database. And SQL is a basic knowledge for understanding how to organize those data. Hence, this project has helped us organize data in database through python.

REFERENCES

- <https://www.tutorialspoint.com/python/index.htm>
- <https://stackoverflow.com/>
- <https://mail.python.org/>
- <https://docs.python.org/3/library/tkinter.ttk.html>
- <https://www.edureka.co/blog/web-scraping-with-python/>
- <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>