

CENTRAL HINDU BOYS SCHOOL



Project for CBSE
Term- 2 Practical Examination 2021-22

Topic- GUI for MySQL

SUBMITTED TO :

*Anik Kumar Pandey
(PGT Computer)*

SUBMITTED BY :

*Name: Rajaneesh Kumar
Class: XII A1
Roll no: 23729103
Subject: Computer Project
Session: 2021- 2022*

Certificate

This is to certify that this project is a bonafide work done by Rajaneesh Kumar of class XII A1 (Maths) session 2021-22 in a partial fulfilment of CBSE Term- 1 practical Examination 2021-22 and has been carried out under my direct supervision and guidance.

Mr. Anik Kumar Pandey
PGT (Computer Science)

Acknowledgement

I would like to express my special thanks of gratitude to my teacher Anik Pandey sir as well as our principal Niru Wahal Ma'am, who gave me the golden opportunity to do this wonderful project on the topic GUI for MySQL which also helped me in doing a lot of research and I came to know about so many new things I am really thankful to them.

Secondly, I would also like to thank my parents and friends who helped me a lot in finalizing this project within limited time frame.

Last but not least I would like to thank all those who had helped
(Directly or indirectly) towards the completion of this project.

Class – XII (Maths)

Index

S. No.	Contents	Page No.
1.	Certificate	2
2.	Acknowledgment	3
3.	Current Manual system	5
4.	Proposed automated system	6
5.	Introduction about Project	7
6.	Objective	8
7.	Requirements	9
8.	System development cycle	10-12
9.	Code	13-20
10.	Output	21-22
11.	Limitations and further Improvements	23
12.	Conclusion	24
13.	Bibliography	25

Current Manual System

A manual system is bookkeeping system where records are maintained by hands without using a computer system or any automatic system. In this system transactions are written in journals, from which the information is manually retrieved into a set of financial statements. These systems suffer from higher rate of inaccuracy and they are much slower than the computerized systems. Manual systems are commonly found in small enterprises that have fewer transactions.

Drawback Of the Manual System

- Lack of security
- Time consuming
- Insufficient storage space
- Difficulties in modifications of data
- Increases cost

Proposed Automated System

Automated system is a combination of both software and hardware which is designed and programmed to work automatically without the need of any human operator to provide input and instructions for each operation.

Automated system allows you to monitor your processes in real time and identify problems as they arrive, enabling quick adjustments along the way. While manual systems can be difficult to coordinate similar to the old cliché that “the right hand doesn’t know what the left hand is doing,” automated systems work in tandem on their own.

Benefits of the Automated System

- Lower operating costs
- Improved worker safety
- Higher output and increased productivity
- Consistency of product quality
- Time efficient

Introduction about Project

This is a project on the topic “GUI for MySQL”. GUI stands for Graphical User Interface it means we are able to manage MySQL Databases by using Buttons and Entry boxes we don't have to enter those complicated MySQL commands.

We can use this project to display the list of tables and the contents of those tables. We can also run MySQL commands in case that feature is not available in the program.

We can also perform four operations on the tables they are:

- ✚ Inserting a new row in a table
- ✚ Updating some data in a table
- ✚ Deleting a row in a table
- ✚ Searching for a row in a table

Objective

The primary aim of the automated system is to reduce human intervention. A human operator (manual system) is prone to errors and fatigue which can lead to a variety of problems. Adapting an automated system will produce substantial benefits on profit, production rates, safety and quality.

The main objectives of this project are:

- ✦ More consistent production
- ✦ Increased repeatability
- ✦ Simplifying operating environment
- ✦ Reduce timework
- ✦ Improving stability and reliability
- ✦ Reduction of mistakes
- ✦ Reducing the operational costs

Requirements

Hardware

- ❖ Processor: Intel(R) Core(TM) i3-6100 CPU @ 3.70GHz
- ❖ Processor speed: 3.70GHz, 3696 MHz, 2 Core(s), 4 Logical Processor(s)
- ❖ Ram: 2 GB or more
- ❖ Hard disk : 2:00 GB

Software:

- ❖ Operating system: windows 7 or above
- ❖ IDE: IDLE Python
- ❖ Front End: Python 3.6 or above
- ❖ Back End: MySQL server 5.0 or above

System Development Cycle

SDLC is a step-by-step procedure or systematic approach to develop software and it is followed within a software organization. It consists of various phase which describe how to design, develop, enhance and maintain particular software.

Phase 1: requirement collection and analysis:

In this phase mainly focus on gathering the business needs from the customer. It determines the requirements like; what should be input data to the system? Who is going to use the system? What should be the output data by the system? These questions are getting answered during this phase.

Phase 2: Feasibility study:

In this step, we examine the feasibility of the proposed system.

Phase 3: Design:

Design is a blueprint of the application and it helps in specifying hardware and requirements of the system and helps in defining architecture of the system.

Phase 4: Coding:

Once the system design document is ready in this phase developer's starts writing the code using any programming language i.e., they start developing the software.

Generally, task is divided in units or modules and assigned to the developers and this coding phase is the longest phase of SDLC

Phase 5: Testing:

During this phase test engineers may encounter some bugs/defects which need to be sent to developers, the developers fix the bugs and sent back to test engineers for testing.

Phase 6: Installation/Deployment:

Once the product developed, tested and works according to the requirements it is installed/deployed at customer place for their use.

Phase 7: Maintenance:

When the customer starts using the software, they may face some issues and needs to be solved from time-to-time means need to fix those issues, tested and handed over back to the customers as soon as possible, which is done in the maintenance phase.

Code

```
import tkinter as tk
import mysql.connector as sql

root=tk.Tk()
root.title("Login Window")
root.geometry("900x500")

def connection():
    obj=sql.connect(host="localhost",user=e1.get(),passwd=e2.get(),database=e3.get())

    if obj.is_connected():
        l4=tk.Label(root,text="Successfully Connected")
        l4.grid(row=4,column=3)
        cursor=obj.cursor()

    ##Functions for showing the table contents
    def ctab1():

        #Function to insert row in a table
        def insert():
            def insert_row():
                values=eval(evalue.get())
                sql="INSERT INTO "+data[0][0]+" VALUES("
                for i in values:
                    if str(type(i))=="<class 'str'>":
                        sql=sql+"'%s'",
                    else:
                        sql=sql+"%s,"
                sql=sql[0:-1]
                sql=sql+')';
                cursor.execute(sql%(eval(evalue.get())))
                obj.commit()
                msg=tk.Label(wininsert,text="Successfully Inserted",font=("Ubuntu
Mono",15,"bold")).grid(row=3,column=2)

            wininsert=tk.Tk()
            wininsert.focus_force()
            wininsert.title("Insert row")
            lq=tk.Label(wininsert,text="INSERT INTO "+data[0][0]+" VALUES
(",font=("Ubuntu Mono",12)).grid(row=1,column=1)
            evalue=tk.Entry(wininsert,width=30,font=("Ubuntu Mono",12))
            evalue.grid(row=1,column=2)
            lq1=tk.Label(wininsert,text=" );",font=("Ubuntu
Mono",12)).grid(row=1,column=3)
```

```

        insertb=tk.Button(wininsert,text="Insert",font=("Ubuntu
Mono",12),command=insert_row).grid(row=2,column=2)
        wininsert.mainloop()

    def delete():
        delete_row=tk.Tk()
        delete_row.title("Delete row")
        delete_row.focus_force()
        lb1=tk.Label(delete_row,text="DELETE FROM "+data[0][0]+" \nWHERE
",font=("Ubuntu Mono",12)).grid(row=1,column=1)
        cln_name=tk.Entry(delete_row,font=("Ubuntu Mono",12))
        cln_name.grid(row=1,column=2)
        lb2=tk.Label(delete_row,text="=",font=("Ubuntu
Mono",12)).grid(row=1,column=3)
        val1=tk.Entry(delete_row,font=("Ubuntu Mono",12))
        val1.grid(row=1,column=4)

        def delete_Button():
            if str(type(eval(val1.get()))=="<class 'int'>":

                sql="delete from "+data[0][0]+" where "+cln_name.get()+" =
%s;"

                cursor.execute(sql%(eval(val1.get()),""))
            else:
                sql="delete from "+data[0][0]+" where "+cln_name.get()+" =
'%s';"

                cursor.execute(sql%(eval(val1.get()),""))

            obj.commit()
            msg=tk.Label(delete_row,text="Successfully Deleted",font=("Ubuntu
Mono",15,"bold")).grid(row=3,column=1)

        bdel=tk.Button(delete_row,text="Delete",command=delete_Button).grid(row=2,column=2)

        delete_row.mainloop()

    def search():
        searchwin=tk.Tk()
        searchwin.focus_force()
        searchwin.title("Search row")
        lquery=tk.Label(searchwin,text="SELECT * FROM
"+data[0][0],font=("Ubuntu Mono",12)).grid(row=1,column=1)
        lquery1=tk.Label(searchwin,text="WHERE",font=("Ubuntu
Mono",12)).grid(row=2,column=1)
        cln=tk.Entry(searchwin,font=("Ubuntu Mono",12))
        cln.grid(row=2,column=2)
        lquery2=tk.Label(searchwin,text="=",font=("Ubuntu
Mono",12)).grid(row=2,column=3)
        val=tk.Entry(searchwin,font=("Ubuntu Mono",12))
        val.grid(row=2,column=4)

```

```

lquery3=tk.Label(searchwin,text=" ;",font=("Ubuntu
Mono",12)).grid(row=2,column=5)
def search_Button():
    if str(type(eval(val.get()))=="<class 'int'>":

        sql="SELECT * FROM "+data[0][0]+" where "+cln.get()+" = %s;"

        cursor.execute(sql%(eval(val.get()),""))
    else:
        sql="SELECT * FROM "+data[0][0]+" where "+cln.get()+" =
'%s';"

        cursor.execute(sql%(eval(val.get()),""))

    dat=""
    for i in cursor.fetchall():
        dat=dat+str(i)+"\n\n"

    head=tk.Label(searchwin,text=20*"="+"\nSearch
Results",font=("Ubuntu Mono",15,"bold")).grid(row=4,column=1)
    dat=tk.Label(searchwin,text=dat,font=("Ubuntu
Mono",12)).grid(row=5,column=1)

    bs=tk.Button(searchwin,text="Search",font=("Ubuntu
Mono",12),command=search_Button).grid(row=3,column=2)
    searchwin.mainloop()

def update():
    updatewin=tk.Tk()
    updatewin.title("Update")
    updatewin.focus_force()
    queryl0=tk.Label(updatewin,text="UPDATE "+data[0][0],font=("Ubuntu
Mono",12)).grid(row=1,column=1)
    queryl1=tk.Label(updatewin,text="SET ",font=("Ubuntu
Mono",12)).grid(row=2,column=1)
    clnset=tk.Entry(updatewin,font=("Ubuntu Mono",12))
    clnset.grid(row=2,column=2)
    queryl2=tk.Label(updatewin,text=" = ",font=("Ubuntu
Mono",12)).grid(row=2,column=3)
    valset=tk.Entry(updatewin,font=("Ubuntu Mono",12))
    valset.grid(row=2,column=4)
    queryl3=tk.Label(updatewin,text="WHERE ",font=("Ubuntu
Mono",12)).grid(row=3,column=1)
    clnw=tk.Entry(updatewin,font=("Ubuntu Mono",12))
    clnw.grid(row=3,column=2)
    queryl4=tk.Label(updatewin,text=" = ",font=("Ubuntu
Mono",12)).grid(row=3,column=3)
    valw=tk.Entry(updatewin,font=("Ubuntu Mono",12))
    valw.grid(row=3,column=4)
    queryl5=tk.Label(updatewin,text=" ;",font=("Ubuntu
Mono",12)).grid(row=3,column=5)

def update_button():
    if str(type(eval(valset.get()))=="<class 'int'>":

```



```

        sql="UPDATE "+data[0][0]+" SET "+cInset.get()+" = %s "
    else:
        sql="UPDATE "+data[0][0]+" SET "+cInset.get()+" = '%s' "

    if str(type(eval(valw.get()))=="<class 'int'>":
        sql=sql+"where "+cInw.get()+" = %s;"
    else:
        sql=sql+"where "+cInw.get()+" = '%s';"

    cursor.execute(sql%(eval(valset.get()),eval(valw.get())))
    obj.commit()
    msg=tk.Label(updatewin,text="Successfully Updated",font=("Ubuntu
Mono",12,"bold")).grid(row=5,column=1)

    update_button=tk.Button(updatewin,text="Update",font=("Ubuntu
Mono",12),command=update_button).grid(row=4,column=2)

    updatewin.mainloop()

    dat=""
    cursor.execute("select * from "+data[0][0]+";")
    for i in cursor.fetchall():
        dat=dat+str(i)+"\n"+200*" - "+" \n"
    table=tk.Tk()
    table.focus_force()
    table.title("Contents of Table")
    binset=tk.Button(table,text="Insert row",font=("Ubuntu
Mono",12),command=insert).pack()
    bupdate=tk.Button(table,text="Update",font=("Ubuntu
Mono",12),command=update).pack()
    bdelete=tk.Button(table,text="Delete row",font=("Ubuntu
Mono",12),command=delete).pack()
    bsearch=tk.Button(table,text="Search",font=("Ubuntu
Mono",12),command=search).pack()
    cIn=tk.Label(table,text="\n"+str(cursor.column_names)+"\n"+30*" ----
"+" \n",font=("HP Simplified",11))
    cIn.pack()
    Labd=tk.Label(table,text=dat,font=("HP Simplified",11))
    Labd.pack()
    table.mainloop()

def ctab2():
    #Function to insert row in a table
    def insert():
        def insert_row():
            values=eval(evalue.get())
            sql="INSERT INTO "+data[1][0]+" VALUES("

```



```

        for i in values:
            if str(type(i)) == "<class 'str'>":
                sql=sql+'"%s",'
            else:
                sql=sql+"%s,"
        sql=sql[0:-1]
        sql=sql+');'
        cursor.execute(sql%(eval(evalue.get()))))
        obj.commit()
        msg=tk.Label(wininsert,text="Successfully Inserted",font=("Ubuntu
Mono",15,"bold")).grid(row=3,column=2)

        wininsert=tk.Tk()
        wininsert.focus_force()
        wininsert.title("Insert row")
        lq=tk.Label(wininsert,text="INSERT INTO "+data[1][0]+" VALUES
(",font=("Ubuntu Mono",12)).grid(row=1,column=1)
        evalue=tk.Entry(wininsert,width=30,font=("Ubuntu Mono",12))
        evalue.grid(row=1,column=2)
        lq1=tk.Label(wininsert,text=" );",font=("Ubuntu
Mono",12)).grid(row=1,column=3)
        insertb=tk.Button(wininsert,text="Insert",font=("Ubuntu
Mono",12),command=insert_row).grid(row=2,column=2)
        wininsert.mainloop()

    def delete():
        delete_row=tk.Tk()
        delete_row.title("Delete row")
        delete_row.focus_force()
        lb1=tk.Label(delete_row,text="DELETE FROM "+data[1][0]+" \nWHERE
",font=("Ubuntu Mono",12)).grid(row=1,column=1)
        cln_name=tk.Entry(delete_row,font=("Ubuntu Mono",12))
        cln_name.grid(row=1,column=2)
        lb2=tk.Label(delete_row,text="=",font=("Ubuntu
Mono",12)).grid(row=1,column=3)
        val1=tk.Entry(delete_row,font=("Ubuntu Mono",12))
        val1.grid(row=1,column=4)

    def delete_Button():
        if str(type(eval(val1.get())))== "<class 'int'>":

            sql="delete from "+data[1][0]+" where "+cln_name.get()+" =
%s;"

            cursor.execute(sql%(eval(val1.get())+","))
        else:
            sql="delete from "+data[1][0]+" where "+cln_name.get()+" =
'%s';"

            cursor.execute(sql%(eval(val1.get())+","))

        obj.commit()
        msg=tk.Label(delete_row,text="Successfully Deleted",font=("Ubuntu
Mono",15,"bold")).grid(row=3,column=1)

```

```

bdel=tk.Button(delete_row,text="Delete",command=delete_Button).grid(row=2,column=2)

delete_row.mainloop()

def search():
    searchwin=tk.Tk()
    searchwin.focus_force()
    searchwin.title("Search row")
    lquery=tk.Label(searchwin,text="SELECT * FROM
"+data[1][0],font=("Ubuntu Mono",12)).grid(row=1,column=1)
    lquery1=tk.Label(searchwin,text="WHERE",font=("Ubuntu
Mono",12)).grid(row=2,column=1)
    cln=tk.Entry(searchwin,font=("Ubuntu Mono",12))
    cln.grid(row=2,column=2)
    lquery2=tk.Label(searchwin,text="=",font=("Ubuntu
Mono",12)).grid(row=2,column=3)
    val=tk.Entry(searchwin,font=("Ubuntu Mono",12))
    val.grid(row=2,column=4)
    lquery3=tk.Label(searchwin,text=";",font=("Ubuntu
Mono",12)).grid(row=2,column=5)
    def search_Button():
        if str(type(eval(val.get()))=="<class 'int'>"):

            sql="SELECT * FROM "+data[1][0]+" where "+cln.get()+" = %s;"

            cursor.execute(sql%(eval(val.get()),"))
        else:
            sql="SELECT * FROM "+data[1][0]+" where "+cln.get()+" =
'%s';"

            cursor.execute(sql%(eval(val.get()),"))

        dat=""
        for i in cursor.fetchall():
            dat=dat+str(i)+"\n\n"

        head=tk.Label(searchwin,text=20*"="+"\nSearch
Results",font=("Ubuntu Mono",15,"bold")).grid(row=4,column=1)
        dat=tk.Label(searchwin,text=dat,font=("Ubuntu
Mono",12)).grid(row=5,column=1)

        bs=tk.Button(searchwin,text="Search",font=("Ubuntu
Mono",12),command=search_Button).grid(row=3,column=2)
        searchwin.mainloop()

def update():
    updatewin=tk.Tk()
    updatewin.title("Update")
    updatewin.focus_force()
    queryl0=tk.Label(updatewin,text="UPDATE "+data[1][0],font=("Ubuntu
Mono",12)).grid(row=1,column=1)

```

```

        queryl1=tk.Label(updatewin,text="SET ",font=("Ubuntu
Mono",12)).grid(row=2,column=1)
        clnset=tk.Entry(updatewin,font=("Ubuntu Mono",12))
        clnset.grid(row=2,column=2)
        queryl2=tk.Label(updatewin,text=" = ",font=("Ubuntu
Mono",12)).grid(row=2,column=3)
        valset=tk.Entry(updatewin,font=("Ubuntu Mono",12))
        valset.grid(row=2,column=4)
        queryl3=tk.Label(updatewin,text="WHERE ",font=("Ubuntu
Mono",12)).grid(row=3,column=1)
        clnw=tk.Entry(updatewin,font=("Ubuntu Mono",12))
        clnw.grid(row=3,column=2)
        queryl4=tk.Label(updatewin,text=" = ",font=("Ubuntu
Mono",12)).grid(row=3,column=3)
        valw=tk.Entry(updatewin,font=("Ubuntu Mono",12))
        valw.grid(row=3,column=4)
        queryl5=tk.Label(updatewin,text=" ;",font=("Ubuntu
Mono",12)).grid(row=3,column=5)

    def update_button():
        if str(type(eval(valset.get()))=="<class 'int'>"):

            sql="UPDATE "+data[1][0]+" SET "+clnset.get()+" = %s "

        else:
            sql="UPDATE "+data[1][0]+" SET "+clnset.get()+" = '%s' "

        if str(type(eval(valw.get()))=="<class 'int'>"):

            sql=sql+"where "+clnw.get()+" = %s;"

        else:
            sql=sql+"where "+clnw.get()+" = '%s';"

        cursor.execute(sql%(eval(valset.get()),eval(valw.get())))
        obj.commit()
        msg=tk.Label(updatewin,text="Successfully Updated",font=("Ubuntu
Mono",12,"bold")).grid(row=5,column=1)

        update_button=tk.Button(updatewin,text="Update",font=("Ubuntu
Mono",12),command=update_button).grid(row=4,column=2)

    updatewin.mainloop()

    dat=""
    cursor.execute("select * from "+data[1][0]+";")
    for i in cursor.fetchall():
        dat=dat+str(i)+"\n"+200*" - "+" \n"
    table=tk.Tk()
    table.focus_force()
    table.title("Contents of Table")

```

```

        bininsert=tk.Button(table,text="Insert row",font=("Ubuntu
Mono",12),command=insert).pack()
        bupdate=tk.Button(table,text="Update",font=("Ubuntu
Mono",12),command=update).pack()
        bdelete=tk.Button(table,text="Delete row",font=("Ubuntu
Mono",12),command=delete).pack()
        bsearch=tk.Button(table,text="Search",font=("Ubuntu
Mono",12),command=search).pack()
        cln=tk.Label(table,text="\n"+str(cursor.column_names)+"\n"+30*"----
"+"+\n",font=("HP Simplified",11))
        cln.pack()
        Labd=tk.Label(table,text=dat,font=("HP Simplified",11))
        Labd.pack()
        table.mainloop()

def ctab3():
    #Function to insert row in a table
    def insert():
        def insert_row():
            values=eval(evalue.get())
            sql="INSERT INTO "+data[2][0]+" VALUES("
            for i in values:
                if str(type(i))=="<class 'str'>":
                    sql=sql+'"%s",'
                else:
                    sql=sql+'%s,'
            sql=sql[:-1]
            sql=sql+')';
            cursor.execute(sql%(eval(evalue.get())))
            obj.commit()
            msg=tk.Label(wininsert,text="Successfully Inserted",font=("Ubuntu
Mono",15,"bold")).grid(row=3,column=2)

            wininsert=tk.Tk()
            wininsert.focus_force()
            wininsert.title("Insert row")
            lq=tk.Label(wininsert,text="INSERT INTO "+data[2][0]+" VALUES
(",font=("Ubuntu Mono",12)).grid(row=1,column=1)
            evalue=tk.Entry(wininsert,width=30,font=("Ubuntu Mono",12))
            evalue.grid(row=1,column=2)
            lq1=tk.Label(wininsert,text=" );",font=("Ubuntu
Mono",12)).grid(row=1,column=3)
            insertb=tk.Button(wininsert,text="Insert",font=("Ubuntu
Mono",12),command=insert_row).grid(row=2,column=2)
            wininsert.mainloop()

        def delete():
            delete_row=tk.Tk()
            delete_row.title("Delete row")
            delete_row.focus_force()
            lb1=tk.Label(delete_row,text="DELETE FROM "+data[2][0]+" \nWHERE
",font=("Ubuntu Mono",12)).grid(row=1,column=1)

```

```

        cln_name=tk.Entry(delete_row,font=("Ubuntu Mono",12))
        cln_name.grid(row=1,column=2)
        lb2=tk.Label(delete_row,text="",font=("Ubuntu
Mono",12)).grid(row=1,column=3)
        val1=tk.Entry(delete_row,font=("Ubuntu Mono",12))
        val1.grid(row=1,column=4)

        def delete_Button():
            if str(type(eval(val1.get())))==<class 'int'>:

                sql="delete from "+data[2][0]+" where "+cln_name.get()+" =
%s;"

                cursor.execute(sql%(eval(val1.get()),""))
            else:
                sql="delete from "+data[2][0]+" where "+cln_name.get()+" =
'%s';"

                cursor.execute(sql%(eval(val1.get()),""))

            obj.commit()
            msg=tk.Label(delete_row,text="Successfully Deleted",font=("Ubuntu
Mono",15,"bold")).grid(row=3,column=1)

        bdel=tk.Button(delete_row,text="Delete",command=delete_Button).grid(row=2,column=2)

        delete_row.mainloop()

        def search():
            searchwin=tk.Tk()
            searchwin.focus_force()
            searchwin.title("Search row")
            lquery=tk.Label(searchwin,text="SELECT * FROM
"+data[2][0],font=("Ubuntu Mono",12)).grid(row=1,column=1)
            lquery1=tk.Label(searchwin,text="WHERE",font=("Ubuntu
Mono",12)).grid(row=2,column=1)
            cln=tk.Entry(searchwin,font=("Ubuntu Mono",12))
            cln.grid(row=2,column=2)
            lquery2=tk.Label(searchwin,text="",font=("Ubuntu
Mono",12)).grid(row=2,column=3)
            val=tk.Entry(searchwin,font=("Ubuntu Mono",12))
            val.grid(row=2,column=4)
            lquery3=tk.Label(searchwin,text=";",font=("Ubuntu
Mono",12)).grid(row=2,column=5)
            def search_Button():
                if str(type(eval(val.get())))==<class 'int'>:

                    sql="SELECT * FROM "+data[2][0]+" where "+cln.get()+" = %s;"

                    cursor.execute(sql%(eval(val.get()),""))
                else:
                    sql="SELECT * FROM "+data[2][0]+" where "+cln.get()+" =
'%s';"

```

```

        cursor.execute(sql%(eval(val.get()+",")))

        dat=""
        for i in cursor.fetchall():
            dat=dat+str(i)+"\n\n"

        head=tk.Label(searchwin,text=20*"="+"\nSearch
Results",font=("Ubuntu Mono",15,"bold")).grid(row=4,column=1)
        dat=tk.Label(searchwin,text=dat,font=("Ubuntu
Mono",12)).grid(row=5,column=1)

        bs=tk.Button(searchwin,text="Search",font=("Ubuntu
Mono",12),command=search_Button).grid(row=3,column=2)
        searchwin.mainloop()

    def update():
        updatewin=tk.Tk()
        updatewin.title("Update")
        updatewin.focus_force()
        queryl0=tk.Label(updatewin,text="UPDATE "+data[2][0],font=("Ubuntu
Mono",12)).grid(row=1,column=1)
        queryl1=tk.Label(updatewin,text="SET ",font=("Ubuntu
Mono",12)).grid(row=2,column=1)
        clnset=tk.Entry(updatewin,font=("Ubuntu Mono",12))
        clnset.grid(row=2,column=2)
        queryl2=tk.Label(updatewin,text=" = ",font=("Ubuntu
Mono",12)).grid(row=2,column=3)
        valset=tk.Entry(updatewin,font=("Ubuntu Mono",12))
        valset.grid(row=2,column=4)
        queryl3=tk.Label(updatewin,text="WHERE ",font=("Ubuntu
Mono",12)).grid(row=3,column=1)
        clnw=tk.Entry(updatewin,font=("Ubuntu Mono",12))
        clnw.grid(row=3,column=2)
        queryl4=tk.Label(updatewin,text=" = ",font=("Ubuntu
Mono",12)).grid(row=3,column=3)
        valw=tk.Entry(updatewin,font=("Ubuntu Mono",12))
        valw.grid(row=3,column=4)
        queryl5=tk.Label(updatewin,text=" ;",font=("Ubuntu
Mono",12)).grid(row=3,column=5)

    def update_button():
        if str(type(eval(valset.get()))=="<class 'int'>":

            sql="UPDATE "+data[2][0]+" SET "+clnset.get()+" = %s "

        else:
            sql="UPDATE "+data[2][0]+" SET "+clnset.get()+" = '%s' "

        if str(type(eval(valw.get()))=="<class 'int'>":

            sql=sql+"where "+clnw.get()+" = %s;"

```



```

        else:
            sql=sql+"where "+clnw.get()+" = '%s';"

            cursor.execute(sql%(eval(valset.get()),eval(valw.get()))))
            obj.commit()
            msg=tk.Label(updatewin,text="Successfully Updated",font=("Ubuntu
Mono",12,"bold")).grid(row=5,column=1)

            update_button=tk.Button(updatewin,text="Update",font=("Ubuntu
Mono",12),command=update_button).grid(row=4,column=2)

            updatewin.mainloop()

            dat=""
            cursor.execute("select * from "+data[2][0]+";")
            for i in cursor.fetchall():
                dat=dat+str(i)+"\n"+200*" - "+" \n"
            table=tk.Tk()
            table.focus_force()
            table.title("Contents of Table")
            binsert=tk.Button(table,text="Insert row",font=("Ubuntu
Mono",12),command=insert).pack()
            bupdate=tk.Button(table,text="Update",font=("Ubuntu
Mono",12),command=update).pack()
            bdelete=tk.Button(table,text="Delete row",font=("Ubuntu
Mono",12),command=delete).pack()
            bsearch=tk.Button(table,text="Search",font=("Ubuntu
Mono",12),command=search).pack()
            cln=tk.Label(table,text="\n"+str(cursor.column_names)+"\n"+30*" ----
"+" \n",font=("HP Simplified",11))
            cln.pack()
            Labd=tk.Label(table,text=dat,font=("HP Simplified",11))
            Labd.pack()
            table.mainloop()

def ctab4():
    #Function to insert row in a table
    def insert():
        def insert_row():
            values=eval(evalue.get())
            sql="INSERT INTO "+data[3][0]+" VALUES("
            for i in values:
                if str(type(i))=="<class 'str'>":
                    sql=sql+'"%s",'
                else:
                    sql=sql+'%s,'
            sql=sql[0:-1]
            sql=sql+')';
            cursor.execute(sql%(eval(evalue.get()))))
            obj.commit()

```

```

        msg=tk.Label(wininsert,text="Successfully Inserted",font=("Ubuntu
Mono",15,"bold")).grid(row=3,column=2)

        wininsert=tk.Tk()
        wininsert.focus_force()
        wininsert.title("Insert row")
        lq=tk.Label(wininsert,text="INSERT INTO "+data[3][0]+" VALUES
(",font=("Ubuntu Mono",12)).grid(row=1,column=1)
        evalue=tk.Entry(wininsert,width=30,font=("Ubuntu Mono",12))
        evalue.grid(row=1,column=2)
        lq1=tk.Label(wininsert,text=" );",font=("Ubuntu
Mono",12)).grid(row=1,column=3)
        insertb=tk.Button(wininsert,text="Insert",font=("Ubuntu
Mono",12),command=insert_row).grid(row=2,column=2)
        wininsert.mainloop()

    def delete():
        delete_row=tk.Tk()
        delete_row.title("Delete row")
        delete_row.focus_force()
        lb1=tk.Label(delete_row,text="DELETE FROM "+data[3][0]+" \nWHERE
",font=("Ubuntu Mono",12)).grid(row=1,column=1)
        cln_name=tk.Entry(delete_row,font=("Ubuntu Mono",12))
        cln_name.grid(row=1,column=2)
        lb2=tk.Label(delete_row,text="=",font=("Ubuntu
Mono",12)).grid(row=1,column=3)
        val1=tk.Entry(delete_row,font=("Ubuntu Mono",12))
        val1.grid(row=1,column=4)

        def delete_Button():
            if str(type(eval(val1.get()))=="<class 'int'>":

                sql="delete from "+data[3][0]+" where "+cln_name.get()+" =
%s;"

                cursor.execute(sql%(eval(val1.get()),""))
            else:
                sql="delete from "+data[3][0]+" where "+cln_name.get()+" =
'%s';"

                cursor.execute(sql%(eval(val1.get()),""))

            obj.commit()
            msg=tk.Label(delete_row,text="Successfully Deleted",font=("Ubuntu
Mono",15,"bold")).grid(row=3,column=1)

        bdel=tk.Button(delete_row,text="Delete",command=delete_Button).grid(row=2,column=2)

        delete_row.mainloop()

    def search():
        searchwin=tk.Tk()
        searchwin.focus_force()

```



```

        searchwin.title("Search row")
        lquery=tk.Label(searchwin,text="SELECT * FROM
"+data[3][0],font=("Ubuntu Mono",12)).grid(row=1,column=1)
        lquery1=tk.Label(searchwin,text="WHERE",font=("Ubuntu
Mono",12)).grid(row=2,column=1)
        cln=tk.Entry(searchwin,font=("Ubuntu Mono",12))
        cln.grid(row=2,column=2)
        lquery2=tk.Label(searchwin,text="=",font=("Ubuntu
Mono",12)).grid(row=2,column=3)
        val=tk.Entry(searchwin,font=("Ubuntu Mono",12))
        val.grid(row=2,column=4)
        lquery3=tk.Label(searchwin,text=";",font=("Ubuntu
Mono",12)).grid(row=2,column=5)
        def search_Button():
            if str(type(eval(val.get()))=="<class 'int'>":

                sql="SELECT * FROM "+data[3][0]+" where "+cln.get()+" = %s;"

                cursor.execute(sql%(eval(val.get()),""))
            else:
                sql="SELECT * FROM "+data[3][0]+" where "+cln.get()+" =
'%s';"

                cursor.execute(sql%(eval(val.get()),""))

            dat=""
            for i in cursor.fetchall():
                dat=dat+str(i)+"\n\n"

            head=tk.Label(searchwin,text=20*"="+"\nSearch
Results",font=("Ubuntu Mono",15,"bold")).grid(row=4,column=1)
            dat=tk.Label(searchwin,text=dat,font=("Ubuntu
Mono",12)).grid(row=5,column=1)

            bs=tk.Button(searchwin,text="Search",font=("Ubuntu
Mono",12),command=search_Button).grid(row=3,column=2)
            searchwin.mainloop()

        def update():
            updatewin=tk.Tk()
            updatewin.title("Update")
            updatewin.focus_force()
            queryl0=tk.Label(updatewin,text="UPDATE "+data[3][0],font=("Ubuntu
Mono",12)).grid(row=1,column=1)
            queryl1=tk.Label(updatewin,text="SET ",font=("Ubuntu
Mono",12)).grid(row=2,column=1)
            clnset=tk.Entry(updatewin,font=("Ubuntu Mono",12))
            clnset.grid(row=2,column=2)
            queryl2=tk.Label(updatewin,text=" = ",font=("Ubuntu
Mono",12)).grid(row=2,column=3)
            valset=tk.Entry(updatewin,font=("Ubuntu Mono",12))
            valset.grid(row=2,column=4)
            queryl3=tk.Label(updatewin,text="WHERE ",font=("Ubuntu
Mono",12)).grid(row=3,column=1)

```

```

        clnw=tk.Entry(updatewin,font=("Ubuntu Mono",12))
        clnw.grid(row=3,column=2)
        queryl4=tk.Label(updatewin,text=" = ",font=("Ubuntu
Mono",12)).grid(row=3,column=3)
        valw=tk.Entry(updatewin,font=("Ubuntu Mono",12))
        valw.grid(row=3,column=4)
        queryl5=tk.Label(updatewin,text=" ;",font=("Ubuntu
Mono",12)).grid(row=3,column=5)

    def update_button():
        if str(type(eval(valset.get()))=="<class 'int'>":

            sql="UPDATE "+data[3][0]+" SET "+clnset.get()+" = %s "

        else:
            sql="UPDATE "+data[3][0]+" SET "+clnset.get()+" = '%s' "

        if str(type(eval(valw.get()))=="<class 'int'>":

            sql=sql+"where "+clnw.get()+" = %s;"

        else:
            sql=sql+"where "+clnw.get()+" = '%s';"

        cursor.execute(sql%(eval(valset.get()),eval(valw.get())))
        obj.commit()
        msg=tk.Label(updatewin,text="Successfully Updated",font=("Ubuntu
Mono",12,"bold")).grid(row=5,column=1)

        update_button=tk.Button(updatewin,text="Update",font=("Ubuntu
Mono",12),command=update_button).grid(row=4,column=2)

        updatewin.mainloop()

    dat=""
    cursor.execute("select * from "+data[3][0]+";")
    for i in cursor.fetchall():
        dat=dat+str(i)+"\n"+200*" - "+" \n"
    table=tk.Tk()
    table.focus_force()
    table.title("Contents of Table")
    binset=tk.Button(table,text="Insert row",font=("Ubuntu
Mono",12),command=insert).pack()
    bupdate=tk.Button(table,text="Update",font=("Ubuntu
Mono",12),command=update).pack()
    bdelete=tk.Button(table,text="Delete row",font=("Ubuntu
Mono",12),command=delete).pack()
    bsearch=tk.Button(table,text="Search",font=("Ubuntu
Mono",12),command=search).pack()
    cln=tk.Label(table,text="\n"+str(cursor.column_names)+"\n"+30*" ----
+"+" \n",font=("HP Simplified",11))

```

```

        cln.pack()
        Labd=tk.Label(table,text=dat,font=("HP Simplified",11))
        Labd.pack()
        table.mainloop()

def ctab5():
    #Function to insert row in a table
    def insert():
        def insert_row():
            values=eval(evalue.get())
            sql="INSERT INTO "+data[4][0]+" VALUES("
            for i in values:
                if str(type(i))=="<class 'str'>":
                    sql=sql+'"%s",'
                else:
                    sql=sql+"%s,"
            sql=sql[0:-1]
            sql=sql+');'
            cursor.execute(sql%(eval(evalue.get())))
            obj.commit()
            msg=tk.Label(winsert,text="Successfully Inserted",font=("Ubuntu
Mono",15,"bold")).grid(row=3,column=2)

            winsert=tk.Tk()
            winsert.focus_force()
            winsert.title("Insert row")
            lq=tk.Label(winsert,text="INSERT INTO "+data[4][0]+" VALUES
(",font=("Ubuntu Mono",12)).grid(row=1,column=1)
            evalue=tk.Entry(winsert,width=30,font=("Ubuntu Mono",12))
            evalue.grid(row=1,column=2)
            lq1=tk.Label(winsert,text=" );",font=("Ubuntu
Mono",12)).grid(row=1,column=3)
            insertb=tk.Button(winsert,text="Insert",font=("Ubuntu
Mono",12),command=insert_row).grid(row=2,column=2)
            winsert.mainloop()

        def delete():
            delete_row=tk.Tk()
            delete_row.title("Delete row")
            delete_row.focus_force()
            lb1=tk.Label(delete_row,text="DELETE FROM "+data[4][0]+" \nWHERE
",font=("Ubuntu Mono",12)).grid(row=1,column=1)
            cln_name=tk.Entry(delete_row,font=("Ubuntu Mono",12))
            cln_name.grid(row=1,column=2)
            lb2=tk.Label(delete_row,text="=",font=("Ubuntu
Mono",12)).grid(row=1,column=3)
            val1=tk.Entry(delete_row,font=("Ubuntu Mono",12))
            val1.grid(row=1,column=4)

            def delete_Button():
                if str(type(eval(val1.get()))=="<class 'int'>":

```

```

        sql="delete from "+data[4][0]+" where "+cln_name.get()+" =
%s;"

        cursor.execute(sql%(eval(val1.get()),""))
    else:
        sql="delete from "+data[4][0]+" where "+cln_name.get()+" =
'%s';"

        cursor.execute(sql%(eval(val1.get()),""))

    obj.commit()
    msg=tk.Label(delete_row,text="Successfully Deleted",font=("Ubuntu
Mono",15,"bold")).grid(row=3,column=1)

bdel=tk.Button(delete_row,text="Delete",command=delete_Button).grid(row=2,column=2)

delete_row.mainloop()

def search():
    searchwin=tk.Tk()
    searchwin.focus_force()
    searchwin.title("Search row")
    lquery=tk.Label(searchwin,text="SELECT * FROM
"+data[4][0],font=("Ubuntu Mono",12)).grid(row=1,column=1)
    lquery1=tk.Label(searchwin,text="WHERE",font=("Ubuntu
Mono",12)).grid(row=2,column=1)
    cln=tk.Entry(searchwin,font=("Ubuntu Mono",12))
    cln.grid(row=2,column=2)
    lquery2=tk.Label(searchwin,text="=",font=("Ubuntu
Mono",12)).grid(row=2,column=3)
    val=tk.Entry(searchwin,font=("Ubuntu Mono",12))
    val.grid(row=2,column=4)
    lquery3=tk.Label(searchwin,text=";",font=("Ubuntu
Mono",12)).grid(row=2,column=5)
    def search_Button():
        if str(type(eval(val.get()))=="<class 'int'>"):

            sql="SELECT * FROM "+data[4][0]+" where "+cln.get()+" = %s;"

            cursor.execute(sql%(eval(val.get()),""))
        else:
            sql="SELECT * FROM "+data[4][0]+" where "+cln.get()+" =
'%s';"

            cursor.execute(sql%(eval(val.get()),""))

        dat=""
        for i in cursor.fetchall():
            dat=dat+str(i)+"\n\n"

        head=tk.Label(searchwin,text=20*" "+"Search
Results",font=("Ubuntu Mono",15,"bold")).grid(row=4,column=1)
        dat=tk.Label(searchwin,text=dat,font=("Ubuntu
Mono",12)).grid(row=5,column=1)

```

```

        bs=tk.Button(searchwin,text="Search",font=("Ubuntu
Mono",12),command=search_Button).grid(row=3,column=2)
        searchwin.mainloop()

    def update():
        updatewin=tk.Tk()
        updatewin.title("Update")
        updatewin.focus_force()
        queryl0=tk.Label(updatewin,text="UPDATE "+data[4][0],font=("Ubuntu
Mono",12)).grid(row=1,column=1)
        queryl1=tk.Label(updatewin,text="SET ",font=("Ubuntu
Mono",12)).grid(row=2,column=1)
        clnset=tk.Entry(updatewin,font=("Ubuntu Mono",12))
        clnset.grid(row=2,column=2)
        queryl2=tk.Label(updatewin,text=" = ",font=("Ubuntu
Mono",12)).grid(row=2,column=3)
        valset=tk.Entry(updatewin,font=("Ubuntu Mono",12))
        valset.grid(row=2,column=4)
        queryl3=tk.Label(updatewin,text="WHERE ",font=("Ubuntu
Mono",12)).grid(row=3,column=1)
        clnw=tk.Entry(updatewin,font=("Ubuntu Mono",12))
        clnw.grid(row=3,column=2)
        queryl4=tk.Label(updatewin,text=" = ",font=("Ubuntu
Mono",12)).grid(row=3,column=3)
        valw=tk.Entry(updatewin,font=("Ubuntu Mono",12))
        valw.grid(row=3,column=4)
        queryl5=tk.Label(updatewin,text=" ;",font=("Ubuntu
Mono",12)).grid(row=3,column=5)

    def update_button():
        if str(type(eval(valset.get())))==<class 'int'>:

            sql="UPDATE "+data[4][0]+" SET "+clnset.get()+" = %s "

        else:
            sql="UPDATE "+data[4][0]+" SET "+clnset.get()+" = '%s' "

        if str(type(eval(valw.get())))==<class 'int'>:

            sql=sql+"where "+clnw.get()+" = %s;"

        else:
            sql=sql+"where "+clnw.get()+" = '%s';"

        cursor.execute(sql%(eval(valset.get()),eval(valw.get())))
        obj.commit()
        msg=tk.Label(updatewin,text="Successfully Updated",font=("Ubuntu
Mono",12,"bold")).grid(row=5,column=1)

```

```

        update_button=tk.Button(updatewin,text="Update",font=("Ubuntu
Mono",12),command=update_button).grid(row=4,column=2)

        updatewin.mainloop()

        dat=""
        cursor.execute("select * from "+data[4][0]+";")
        for i in cursor.fetchall():
            dat=dat+str(i)+"\n"+200*" "+"+\n"
        table=tk.Tk()
        table.focus_force()
        table.title("Contents of Table")
        binser=tk.Button(table,text="Insert row",font=("Ubuntu
Mono",12),command=insert).pack()
        bupdate=tk.Button(table,text="Update",font=("Ubuntu
Mono",12),command=update).pack()
        bdelete=tk.Button(table,text="Delete row",font=("Ubuntu
Mono",12),command=delete).pack()
        bsearch=tk.Button(table,text="Search",font=("Ubuntu
Mono",12),command=search).pack()
        cln=tk.Label(table,text="\n"+str(cursor.column_names)+"\n"+30*"----
"+"+\n",font=("HP Simplified",11))
        cln.pack()
        Labd=tk.Label(table,text=dat,font=("HP Simplified",11))
        Labd.pack()
        table.mainloop()

    cursor.execute("Show tables;")
    data=cursor.fetchall()

    ## Window for showing the list of Tables
    boot=tk.Tk()
    boot.focus_force()
    boot.title("List of Tables")
    boot.geometry("500x400")
    lab1=tk.Label(boot,text="List of Tables : ",font=("HP Simplified",15))
    lab1.pack()
    gap0=tk.Label(boot,text="-----")
    gap0.pack()
    tab1=tk.Button(boot,text=data[0][0],command=ctab1,font=("HP Simplified",15))
    tab1.pack()
    gap1=tk.Label(boot,text="-----")
    gap1.pack()
    tab2=tk.Button(boot,text=data[1][0],command=ctab2,font=("HP Simplified",15))
    tab2.pack()
    gap2=tk.Label(boot,text="-----")
    gap2.pack()
    tab3=tk.Button(boot,text=data[2][0],command=ctab3,font=("HP Simplified",15))
    tab3.pack()
    gap3=tk.Label(boot,text="-----")
    gap3.pack()
    tab4=tk.Button(boot,text=data[3][0],command=ctab4,font=("HP Simplified",15))

```



```

        tab4.pack()
        gap4=tk.Label(boot,text="-----")
        gap4.pack()
        tab5=tk.Button(boot,text=data[4][0],command=ctab5,font=("HP Simplified",15))
        tab5.pack()

        boot.mainloop()
        obj.close()

    else:
        l4=tk.Label(root,text="Failed to Connect",font=("HP Simplified",15))
        l4.grid(row=3,column=3)

def execute():
    dat=""
    obj=sql.connect(host="localhost",user="root",passwd="HP 245G5",database=e3.get())
    cursor=obj.cursor()
    values=(eval(val.get()))
    cursor.execute(cmd.get()%(eval(val.get())))
    for i in cursor.fetchall():
        dat=dat+str(i)+"\n\n"
    table=tk.Tk()
    table.focus_force()
    table.title("Command Output")
    cln=tk.Label(table,text=str(cursor.column_names)+"\n",font=("HP Simplified",15))
    cln.pack()
    Labd=tk.Label(table,text=dat,font=("HP Simplified",15))
    Labd.pack()
    msg=tk.Label(table,text="Successfully Executed",font=("HP Simplified",15))
    msg.pack()
    table.mainloop()
    obj.close()

def commitexe():
    dat=""
    obj=sql.connect(host="localhost",user="root",passwd="HP 245G5",database=e3.get())
    cursor=obj.cursor()
    values=(eval(val.get()))
    cursor.execute(cmd.get()%(eval(val.get())))
    obj.commit()
    for i in cursor.fetchall():
        dat=dat+str(i)+"\n\n"
    table=tk.Tk()
    table.focus_force()
    table.title("Command Output")
    cln=tk.Label(table,text=str(cursor.column_names)+"\n",font=("HP Simplified",15))
    cln.pack()
    Labd=tk.Label(table,text=dat,font=("HP Simplified",15))
    Labd.pack()
    msg=tk.Label(table,text="Successfully Executed",font=("HP Simplified",15))
    msg.pack()
    table.mainloop()
    obj.close()

```

```

#Login Window_ _ _
heading=tk.Label(root,text="GUI",font=("Shrikhand",60)).grid(row=1,column=1)
heading1=tk.Label(root,text="for",font=("Shrikhand",60)).place(x=320,y=3)
heading2=tk.Label(root,text="MySQL",font=("Shrikhand",60)).place(x=500,y=3)

l1=tk.Label(root,text="Enter Username: ",font=("HP Simplified",15))
l1.grid(row=2,column=1)
e1=tk.Entry(root,font=("HP Simplified",15))
e1.grid(row=2,column=2)
l2=tk.Label(root,text="Enter Password: ",font=("HP Simplified",15))
l2.grid(row=3,column=1)
e2=tk.Entry(root,font=("HP Simplified",15))
e2.grid(row=3,column=2)
l3=tk.Label(root,text="Enter Database: ",font=("HP Simplified",15))
l3.grid(row=4,column=1)
e3=tk.Entry(root,font=("HP Simplified",15))
e3.grid(row=4,column=2)
b1=tk.Button(root,text="Connect and show tables",command=connection,font=("HP
Simplified",15))
b1.grid(row=3,column=3)

border=tk.Label(root,text="\n\n===== \n\n",font=("HP
Simplified",15)).grid(row=5,column=1)

lcmd=tk.Label(root,text="Enter your command here :",font=("Ubuntu Mono",12))
lcmd.grid(row=6,column=1)
cmd=tk.Entry(root,font=("Ubuntu Mono",12),width=40)
cmd.grid(row=7,column=1)
lval=tk.Label(root,text="Enter your values here :",font=("Ubuntu Mono",12))
lval.grid(row=8,column=1)
val=tk.Entry(root,font=("Ubuntu Mono",12),width=40)
val.grid(row=9,column=1)
exe=tk.Button(root,text="Execute",command=execute,font=("Ubuntu Mono",12))
exe.grid(row=7,column=2)
execommit=tk.Button(root,text="Execute with commit
function",command=commitexe,font=("Ubuntu Mono",12))
execommit.grid(row=9,column=2)

credit=tk.Label(root,text="Project by :-\nRajaneesh Kumar",font=("Times New
Roman",11)).place(x=760,y=430)

root.mainloop()

```


Output

Login Window

GUI for MySQL

Enter Username:

Enter Password:

Enter Database:

Successfully Connected

=====

Enter your command here :

Enter your values here :

Project by :-
Rajaneesh Kumar

List of Tables

List of Tables :

Contents of Table		
<div> <div>Insert row</div> <div>Update</div> <div>Delete row</div> <div>Search</div> </div>		
	Widgets	Description
	Button	Button is used to place the buttons in the tkinter, 'Button(<root>,<other parms>')
	Canvas	Canvas is used to draw shapes in your GUL, 'Canvas(<root>,<other parms>')
	Checkbutton	Checkbutton is used to create the check buttons in your application. You can select more than one option at a time., 'Checkbutton(<root>,<other parms>')
	Entry	Entry widget is used to create input fiels in GUL, 'Entry(<root>,<other parms>')
	Frame	Frame is used as containers in the tkinter., 'Frame(<root>,<other parms>')
	Label	Label is used to create a single line widgets like text, images, etc., 'Label(<root>,<other parms>')
	Menu	Menu is used to create menus in the GUL, 'Menu(<root>,<other parms>')

Insert row

INSERT INTO widgets VALUES (xt","multi-line text","Text()");

Insert

Successfully Inserted

Update

UPDATE widgets

SET

Syntax

=

oot>,<other parms>")

WHERE

Widgets

=

"Text"

Update

Successfully Updated

Delete row

DELETE FROM widgets

WHERE

widgets

=

"Text"

Delete

Successfully Deleted

Search row

SELECT * FROM widgets

WHERE

widgets

=

"Button"

Search

Search Results

('Button', 'Button is used to place the buttons in the tkinter', 'Button(<root>,<other parms>')

Limitations and further Improvements

Since the whole project is designed to cater Graphical User Interface it lacks many features some of them are creating databases, creating tables etc. So, an improvement can be done to add these important features in the project.

Secondly the tables displayed are not in proper tabular form. So, as an improvement the project can be modified to display the tables in proper tabular form.

Conclusion

This Program has been designed to achieve maximum efficiency and reduce the time taken to manage MySQL Databases using commands. Its UI is very simple, easily understandable and anyone can use it. It is designed to make the database management easier for everyone and for those people who don't know about MySQL commands.

Bibliography

To develop this project many references were used:

1. Computer science with python by Sumita Arora
Dhanpat rai publications.
2. <https://www.google.com>
3. <https://www.python.org>
4. <https://www.mysql.org>
5. <https://www.isqdirectory.com>
6. <https://www.gekforgeeks.org>
7. <https://stackoverflow.com>
8. <https://www.w3schools.com>