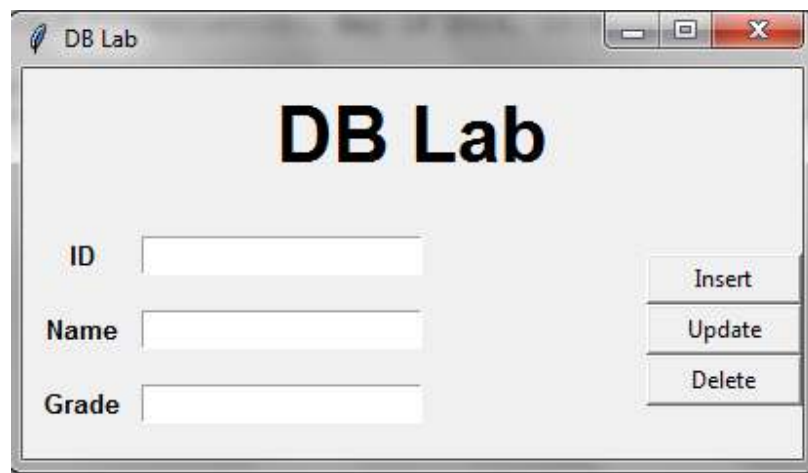


Introduction

A database is a collection of information that is organized so that it can easily be accessed, managed, and updated. In this lab, we will create a python GUI that can handle database queries.

Objectives

- Introduce the student to sqlite3, the self-contained, embedded, full-featured, public-domain, SQL database engine.
 - SQLite is the most used database engine in the world.
- Construct a PC program that can insert and update data to a sqlite database.



Requirements

- Python 3.x (<https://www.python.org/downloads>)
- DB browser (<https://github.com/sqlitebrowser/sqlitebrowser/releases>)

Procedure

- 1- Install python
- 2- Create a frame

```
#import tkinter
from tkinter import *

root =Tk()
```

```
root.geometry("400x200+200+100")
root.title("DB Lab")
```

```
Tops = Frame(root, width=300 , height =100)
Tops.pack(side = TOP)
```

```
f1 = Frame(root, width=300 ,height = 200)
f1.pack(side = LEFT )
```

```
f2 = Frame(root, width=200 ,height = 200)
f2.pack(side = RIGHT)
```

```
lbl = Label(Tops, font=('arial',30,'bold'), text="DB Lab", bd
=10 , anchor = 'w')
lbl.grid(row=0,column = 0)
```

```
lbl1 = Label(f1, font=('arial',10,'bold'), text="ID", bd =10 ,
anchor = 'w')
lbl1.grid(row=0,column = 0)
lbl2 = Label(f1, font=('arial',10,'bold'), text="Name", bd =10 ,
anchor = 'w')
lbl2.grid(row=1,column = 0)
lbl3 = Label(f1, font=('arial',10,'bold'), text="Grade", bd =10 ,
anchor = 'w')
lbl3.grid(row=2,column = 0)
```

```
txt1 = Entry(f1,font=('arial',10,'bold'))
txt1.grid(row=0, column = 1)
```

```
txt2 = Entry(f1,font=('arial',10,'bold'))
txt2.grid(row=1, column = 1)
```

```
txt3 = Entry(f1,font=('arial',10,'bold'))
txt3.grid(row=2, column = 1)
```

```
b1=Button(f2,command=callback1,text="Insert",width =
10).grid(row=0)
b2=Button(f2,command=callback2,text="Update",width =
10).grid(row=1)
```

```
b3=Button(f2,command=callback3,text="Delete",width =
10).grid(row=2)
root.mainloop()
```

3- Create the database

```
import sqlite3
c = conn.cursor()
c.execute("CREATE TABLE IF NOT EXISTS t(id INT
PRIMARY KEY,Name TEXT,grade REAL)")
```

4- Write insert code

```
def callback1():
    try:
        c.execute("INSERT INTO t
VALUES("+txt1.get()+",""+txt2.get()+",""+txt3.get()+")")
        conn.commit()
    except:
        print("INSERT INTO t
VALUES("+txt1.get()+",""+txt2.get()+",""+txt3.get()+")")
        print("Failed to insert")
```

5- Write update code

```
def callback2():
    try:
        c.execute("UPDATE t SET grade = "+txt3.get()+" WHERE
id = "+txt1.get()+"")
        conn.commit()
    except:
        print("UPDATE t SET grade = "+txt3.get()+" WHERE id =
"+txt1.get()+"")
        print("Failed to update")
```

6- Write delete code

```
def callback3():  
    try:  
        c.execute("DELETE FROM t WHERE id =" + txt1.get()+""")  
        conn.commit()  
    except:  
        print("DELETE FROM t WHERE id =" + txt1.get()+""")  
        print("Failed to delete")
```

7- Close the database

```
c.close()  
conn.close()
```

Task

Create the following GUI for a database named inventory that has Name, available, Cost, Total_cost as columns and add a select query based on the name and shown on a table in the bottom of the GUI.

The screenshot shows a GUI application window titled "DB Task". The window has a light gray background and a brown border. At the top center, the title "DB Task" is displayed in a large, bold, black font. Below the title, there are three input fields on the left, each with a label to its left: "Name", "Available", and "Cost". To the right of these input fields, there is a vertical stack of four buttons: "Insert", "Update", "Delete", and "Search by Name". At the bottom of the window, there is a table with four columns: "Name", "Available", "Cost", and "Total Cost". The table is currently empty.