

# Dbms project

---

RA1811003010271 Saarthak  
RA1811003010259 Ajay

# Student name database management system

We made a student database management system to:

1. Insert , new values
2. Update, present values
3. Delete , unwanted values

The data gets saved in a sql file for easier use later

# Tkinter

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps –

- Import the *Tkinter* module.
- Create the GUI application main window.
- Add one or more of the above-mentioned widgets to the GUI application.
- Enter the main event loop to take action against each event triggered by the user.

We use tkinter as our front end interface with buttons and customizable input and different actions namely update, rename, insert

# Sqlite

SQLite3 can be integrated with Python using `sqlite3` module, which was written by Gerhard Haring. It provides an SQL interface compliant with the DB-API 2.0 specification described by PEP 249. You do not need to install this module separately because it is shipped by default along with Python version 2.5.x onwards.

To use `sqlite3` module, you must first create a connection object that represents the database and then optionally you can create a cursor object, which will help you in executing all the SQL statements.

For all our sql application we use `Sqlite3` library in python which helps with the interconnectivity of the database and helps us in the operation componen

```
from tkinter import *
from tkinter.ttk import *
from sqlite3 import *

a=Tk()
a.title('tk')
reg=Label(a,text="REGNO")
regtxt=Entry(a,width=10)
reg.grid(column=0,row=0)
regtxt.grid(column=1,row=0)

name=Label(a,text="NAme")
nametxt=Entry(a,width=10)
name.grid(column=0,row=1)
nametxt.grid(column=1,row=1)

dept=Label(a,text="Dept")
depttxt=Entry(a,width=10)
dept.grid(column=0,row=2)
depttxt.grid(column=1,row=2)

gen=Label(a,text="Gender")
gen.grid(column=0,row=3)
i=IntVar()
ra1=Radiobutton(a,text="MALE",value=1,variable=i)
ra2=Radiobutton(a,text="FEMALE",value=2,variable=i)
se=i.get()
gen=''
```

```

def c():
    se=i.get()

    if(se==1):
        gen='MALE'
    else:
        gen='FEMALE'
    return gen
ra1.grid(column=1,row=3)
ra2.grid(column=2,row=3)

age=Label(a,text="age")
w = Spinbox(a, from_=10, to=100)

age.grid(column=0,row=4)
w.grid(column=1,row=4)

con = connect('myTable.db')
cur = con.cursor()
try:
    cur.execute("""CREATE TABLE student(
        reg INTEGER,
        name VARCHAR(20),
        dept VARCHAR(20),
        gender VARCHAR(20),
        age INTEGER);""")
except OperationalError:
    None

```

```

def insert():
    a=c()
    cur.execute("""INSERT INTO student(reg,name,dept,gender,age) VALUES(?,?,?,?);""",(regtxt.get(),nametxt.get(),depttxt.get(),a,w.get()

b1=Button(a,text='Insert',command=insert)
def update():
    a=c()
    cur.execute("""UPDATE student SET name=?,dept=?,gender=?,age=? WHERE reg=?;""",(nametxt.get(),depttxt.get(),a,w.get(),regtxt.get()))
b2=Button(a,text='Update',command=update)
def delete():
    cur.execute("""DELETE FROM student WHERE reg=?;""",(regtxt.get(),))

b3=Button(a,text='Delete',command=delete)
def select():
    cur.execute("""SELECT * FROM student WHERE reg=?;""",(regtxt.get(),))
    ans=cur.fetchall()
    for i in ans:
        print(ans)

b4=Button(a,text='Select',command=select)

b1.grid(column=0,row=5)
b2.grid(column=1,row=5)
b3.grid(column=0,row=6)
b4.grid(column=1,row=6)

con.commit()
#con.close()

a.mainloop()

```

Output