Python Lab Mini Project

1. Neeraj Kondalkar - 32
2. Abhishek Mohite - 36
3. Prajwal Mudavedkar - 37

Topic: Student Result

What is SQLite ?

SQLite3 is a very easy to use database engine. It is self-contained, serverless, zero-configuration and transactional. It is very fast and lightweight, and the entire database is stored in a single disk file. It is used in a lot of applications as internal data storage. The Python Standard Library includes a module called "sqlite3" intended for working with this database. In order to make any operation with the database we need to get a cursor object and pass the SQL statements to the cursor object to execute them. Finally it is necessary to commit the changes. We are going to create a user table with name, phone, email and password columns. To insert data we use the cursor to execute the query. If you need values from Python variables it is recommended to use the "?" placeholder. Never use string operations or concatenation to make your queries because is very insecure. In this example we are going to insert two users in the database, their information is stored in python variables.

What is GUI and Tkinter ?

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.  
**To 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.

About our project:

The main window consists of 3 options for the user to chose from :

* **INSERT**
* **DISPLAY**
* **QUERY MARKS**
* **DELETE ONE ENTRY**
* **DELETE ALL ENTRIES**

1. **INSERT**

In this window, the user need’s to fill all the required information about the student and on completion click on the submit button.A separate **LIST** will be maintained for each information of the student for storing data in a ordered form. Ex . first\_name\_list , last\_name\_list , gender\_list , etc.

Then the data from the lists will be transferred to the **DICTIONARY** so that all the information about a student will be stored collectively in key: value pair.By using “for” loop for the specified range (length of dictionary) the data is inserted in the database that is created using **SQLITE .**

The data in database is inserted using **FORMAT** function as it has positional formatting property. Format work by putting in one or more replacement fields and placeholders defined by a pair of curly braces **{ }**into a string and calling the **str.format()**.

**CURSORS** are created by the **[connection.cursor()](http://initd.org/psycopg/docs/connection.html" \l "connection.cursor" \o "connection.cursor)** method: they are bound to the connection for the entire lifetime and all the commands are executed in the context of the database session wrapped by the connection.

1. **DISPLAY**

In this window **SCROLLEDTEXT** widget is used to display the Student roll no, Name and Gender of all the students in the database .The data is retrieved from database using FETCHALL method.

1. **QUERY MARKS**

Firstly in this window roll number of student is asked to the user.Then the roll number is then searched in the database which will ultimately display the information & marks of the respective student on the scrolled text window.

1. **DELETE ONE ENTRY**

The user will enter the roll no of the student which he wishes to delete from the database.As the roll no will be unique for each student it will be searched throughoutthe database using database queries.The record will then be deleted from the database.

1. **DELETE ALL ENTRIES**

By executing this function DROP command will be performed in the database.DROP is used to delete a whole database or just a table.The DROP statement destroys the objects like an existing database, table, index, or view.

**Python Program :**

import sqlite3 as sql

from tkinter import \*

from tkinter import scrolledtext

from tkinter import messagebox

# insert entries

def insert():

insert\_window = Tk()

insert\_window.title("Insert")

insert\_window.geometry('300x300')

first\_name\_lbl = Label(insert\_window, text=" first name ")

first\_name\_lbl.grid(column=0, row=0)

first\_name\_txt = Entry(insert\_window, width=10)

first\_name\_txt.grid(column=1, row=0)

last\_name\_lbl = Label(insert\_window, text=" last name ")

last\_name\_lbl.grid(column=0, row=1)

last\_name\_txt = Entry(insert\_window, width=10)

last\_name\_txt.grid(column=1, row=1)

gender\_lbl = Label(insert\_window, text=" gender ")

gender\_lbl.grid(column=0, row=2)

gender\_txt = Entry(insert\_window, width=10)

gender\_txt.grid(column=1, row=2)

maths\_lbl = Label(insert\_window, text=" maths ")

maths\_lbl.grid(column=0, row=3)

maths\_txt = Entry(insert\_window, width=10)

maths\_txt.grid(column=1, row=3)

dsa\_lbl = Label(insert\_window, text=" dsa ")

dsa\_lbl.grid(column=0, row=4)

dsa\_txt = Entry(insert\_window, width=10)

dsa\_txt.grid(column=1, row=4)

db\_lbl = Label(insert\_window, text=" dbms ")

db\_lbl.grid(column=0, row=5)

db\_txt = Entry(insert\_window, width=10)

db\_txt.grid(column=1, row=5)

ld\_lbl = Label(insert\_window, text=" LD ")

ld\_lbl.grid(column=0, row=6)

ld\_txt = Entry(insert\_window, width=10)

ld\_txt.grid(column=1, row=6)

pce\_lbl = Label(insert\_window, text=" PCE ")

pce\_lbl.grid(column=0, row=7)

pce\_txt = Entry(insert\_window, width=10)

pce\_txt.grid(column=1, row=7)

def submit():

sql\_command = """

CREATE TABLE if not exists student(

student\_id INTEGER PRIMARY KEY,

fname VARCHAR(30),

lname VARCHAR(30),

gender CHAR(1),

maths CHAR(2),

dsa CHAR(2),

dbms CHAR(2),

ld CHAR(2),

pce CHAR(2)

);

"""

# create table if not exists

cursor.execute(sql\_command)

first\_name = first\_name\_txt.get()

last\_name = last\_name\_txt.get()

gender = gender\_txt.get()

maths = maths\_txt.get()

dsa = dsa\_txt.get()

db = db\_txt.get()

ld = ld\_txt.get()

pce = pce\_txt.get()

fn\_temp\_list = []

ln\_temp\_list = []

gn\_temp\_list = []

m\_temp\_list = []

ds\_temp\_list = []

db\_temp\_list = []

l\_temp\_list = []

p\_temp\_list = []

dict\_list\_data = {"fname": fn\_temp\_list,

"lname": ln\_temp\_list,

"gender": gn\_temp\_list,

"maths": m\_temp\_list,

"dsa": ds\_temp\_list,

"dbms": db\_temp\_list,

"ld": l\_temp\_list,

"pce": p\_temp\_list

}

fn\_temp\_list.append(str(first\_name))

ln\_temp\_list.append(str(last\_name))

gn\_temp\_list.append(str(gender))

m\_temp\_list.append(str(maths))

ds\_temp\_list.append(str(dsa))

db\_temp\_list.append(str(db))

l\_temp\_list.append(str(ld))

p\_temp\_list.append(str(pce))

dict\_list\_data["fname"] = fn\_temp\_list

dict\_list\_data["lname"] = ln\_temp\_list

dict\_list\_data["gender"] = gn\_temp\_list

dict\_list\_data["maths"] = m\_temp\_list

dict\_list\_data["dsa"] = ds\_temp\_list

dict\_list\_data["dbms"] = db\_temp\_list

dict\_list\_data["ld"] = l\_temp\_list

dict\_list\_data["pce"] = p\_temp\_list

length = int(len(dict\_list\_data["fname"]))

for e in range(length):

format\_str = """

INSERT INTO student (student\_id,

fname,

lname,

gender,

maths,

dsa,

dbms,

ld,

pce)

VALUES (NULL,

"{fname}",

"{lname}",

"{gender}",

"{maths}",

"{dsa}",

"{dbms}",

"{ld}",

"{pce}"

);

"""

sql\_command = format\_str.format(fname=dict\_list\_data["fname"][e],

lname=dict\_list\_data["lname"][e], gender=dict\_list\_data["gender"][e], maths=dict\_list\_data["maths"][e], dsa=dict\_list\_data["dsa"][e], dbms=dict\_list\_data["dbms"][e], ld=dict\_list\_data["ld"][e], pce=dict\_list\_data["pce"][e]

)

cursor.execute(sql\_command)

connection.commit()

# INSERT ONE MORE BUTTON to SUBMIT

submit\_button = Button(insert\_window, text="Submit", command=submit)

submit\_button.grid(column=1, row=8)

back\_button = Button(insert\_window, text="Back", command=insert\_window.destroy)

back\_button.grid(column=1, row=10)

insert\_window.mainloop()

# display all entries

def display\_all():

out\_window = Tk()

out\_window.title("Displaying result")

out\_window.geometry('450x300')

out\_text = scrolledtext.ScrolledText(out\_window, width=50, height=30)

out\_text.grid(column=0, row=0)

cursor.execute("""SELECT \*

FROM student ;""")

for r in cursor:

roll = str(r[0])

name = str(r[1] + " " + r[2])

gender = str(r[3])

roll\_format = "\nStudent roll number : {} ".format(roll)

name\_format = "\n Name : {} ".format(name)

gender\_format = "\n Gender: {} ".format(gender)

out\_text.insert(END, roll\_format)

out\_text.insert(END, name\_format)

out\_text.insert(END, gender\_format)

back\_button = Button(out\_window, text="Back", command=out\_window.destroy)

back\_button.grid(column=1, row=0)

out\_window.mainloop()

# print marks of a student

def query\_marks():

out\_window = Tk()

out\_window.title("Displaying result")

out\_window.geometry('200x100')

roll\_lbl = Label(out\_window, text=" roll no ")

roll\_lbl.grid(column=0, row=0)

roll\_txt = Entry(out\_window, width=10)

roll\_txt.grid(column=1, row=0)

def get\_result():

temp\_roll = roll\_txt.get()

query\_result\_window = Tk()

query\_result\_window.title("displaying result")

query\_result\_window.geometry('300x180')

out\_text = Text(query\_result\_window, width=50, height=50)

out\_text.grid(column=0, row=0)

format\_str = """SELECT student\_id,

fname,

lname,

maths,

dsa,

dbms,

ld,

pce

FROM student

WHERE

"{}"={}

;

"""

sql\_command = format\_str.format(temp\_roll, "student\_id")

cursor.execute(sql\_command)

data = cursor.fetchall()

if len(data) == 0:

messagebox.showinfo("Alert !", "No such entry")

else:

cursor.execute(sql\_command)

for r in cursor:

roll = str(r[0])

fname = str(r[1])

lname = str(r[2])

maths = str(r[3])

dsa = str(r[4])

db = str(r[5])

ld = str(r[6])

pce = str(r[7])

roll\_format = "\nStudent roll number : {} ".format(roll)

fname\_format = "\n first Name : {} ".format(fname)

lname\_format = "\n last Name : {} ".format(lname)

maths\_format = "\n Maths: {} ".format(maths)

dsa\_format = "\n DSA: {} ".format(dsa)

db\_format = "\n DBMS: {} ".format(db)

ld\_format = "\n LD: {} ".format(ld)

pce\_format = "\n PCE: {} ".format(pce)

out\_text.insert(END, roll\_format)

out\_text.insert(END, fname\_format)

out\_text.insert(END, lname\_format)

out\_text.insert(END, maths\_format)

out\_text.insert(END, dsa\_format)

out\_text.insert(END, db\_format)

out\_text.insert(END, ld\_format)

out\_text.insert(END, pce\_format)

#out\_window.mainloop()

get\_button = Button(out\_window, text="Get Result", command=get\_result)

get\_button.grid(column=1, row=1)

back\_button = Button(out\_window, text="Back", command=out\_window.destroy)

back\_button.grid(column=1, row=2)

out\_window.mainloop()

# delete one entry

def delete\_entry():

del\_window = Tk()

del\_window.title("Displaying result")

del\_window.geometry('200x100')

roll\_lbl = Label(del\_window, text=" roll no ")

roll\_lbl.grid(column=0, row=0)

roll\_txt = Entry(del\_window, width=10)

roll\_txt.grid(column=1, row=0)

def delete\_click():

result = messagebox.askquestion("Exit", "Are You Sure?", icon='warning')

if result == "yes":

temp\_roll = roll\_txt.get()

format\_str = """SELECT \*

FROM student WHERE "{}"={};""" # <-EDIT

sql\_command = format\_str.format(temp\_roll, "student\_id")

cursor.execute(sql\_command)

data = cursor.fetchall()

if len(data) == 0:

messagebox.showerror("Error", "No such entry ")

else:

format\_str = """DELETE

FROM student

WHERE

"{}"={}

;

"""

sql\_command = format\_str.format(temp\_roll, "student\_id")

cursor.execute(sql\_command)

connection.commit()

format\_str = "Entry succesfully DELETED for roll number : {}".format(temp\_roll)

messagebox.showinfo("Notification", format\_str)

del\_button = Button(del\_window, text="Delete entry", command=delete\_click)

del\_button.grid(column=1, row=1)

back\_button = Button(del\_window, text="Back", command=del\_window.destroy)

back\_button.grid(column=1, row=2)

del\_window.mainloop()

# delete all entries of student table

def delete\_all():

result = messagebox.askquestion("Delete", "Are You Sure?", icon='warning')

if result=="yes":

cursor.execute("""DROP TABLE student;""")

connection.commit()

messagebox.showinfo("Notification", "DELETED all entries")

def exit\_window():

result = messagebox.askquestion("Exit", "Are You Sure?", icon='warning')

if result == "yes":

window.destroy()

print("[+]Establishing connection to StudentDB")

connection = sql.connect("StudentDB.db")

cursor = connection.cursor()

try:

window = Tk()

window.title("STUDENT RESULT SECTION")

window.geometry('300x300')

insert\_button = Button(window, text="Insert", command=insert)

insert\_button.grid(column=2, row=0)

display\_all\_button = Button(window, text="Display All", command=display\_all)

display\_all\_button.grid(column=2, row=1)

query\_button = Button(window, text="query marks", command=query\_marks)

query\_button.grid(column=2, row=2)

delete\_button = Button(window, text="delete entry", command=delete\_entry)

delete\_button.grid(column=2, row=3)

delete\_all\_button = Button(window, text="delete All", command=delete\_all)

delete\_all\_button.grid(column=2, row=4)

exit\_button = Button(window, text="Exit", command=exit\_window)

exit\_button.grid(column=2, row=5)

window.mainloop()

except KeyboardInterrupt:

print("\n[-]Keyboard Interrupt ")

print("[-]Closing connection to StudentDB")

connection.close()

print("Program successfully terminated")

except sql.OperationalError:

messagebox.showerror("No such table : student")

else:

print("\n[-]Closing connection to StudentDB")

connection.close()

print("Program successfully terminated")

finally:

connection.close()