

Practical : 10

Design the database applications for the following :

- a) **Design a simple database application that stores the records and retrieve the same.**

Explanation : In following example we have created a GUI which will accept the details of the books and it stores it in database on the click of insert button. When we press the show books button it displays the information stored in the database using list box.

Solution :

Step 1 : First we need to create MySQL database and database table which is required for our application as follows :

Create database Library;

Use Library;

```
CREATE TABLE books ( id int primary key auto_increment, title VARCHAR(50),
author VARCHAR(30), publisher VARCHAR(25), year VARCHAR(10),
edition VARCHAR(10));
```

Code :

#10A.py

```
from tkinter import *
```

```
from tkinter import messagebox
```

```
import mysql.connector as mysql
```

```
# 1: creating the connection to our database
```

```
conn = mysql.connect(user='root', password='root',host='127.0.0.1')
```

```
cursor = conn.cursor() # 2: obtaining the cursor
```

```
cursor.execute("USE Library") # 3: Making the database as current
```

```
def insertCall():
```

```
    # 4: creating the SQL statement with values from entry gui
```

```
    sql = "INSERT INTO books (title,author, publisher,year,edition) VALUES
```

```
    (""+bTitle.get()+", ""+aName.get()+", ""+pub.get()+", ""+year.get()+",
```

```
    ""+edi.get()+")"
```

```
    cursor.execute(sql) # 5:Executing the query
```

```
    conn.commit() # 6: Making the changes permanent in the database
```

```
bTitle.delete(0,END) # Clearing the text box
```

```
aName.delete(0,END) # Clearing the text box
```

```
pub.delete(0,END) # Clearing the text box
```

```
year.delete(0,END) # Clearing the text box
```

```
edi.delete(0,END) # Clearing the text box
```

```
bTitle.focus()
```

```
messagebox.showinfo(title='Confirmation', message="Information Inserted.")
```



```
def showBooks():
    sql = "SELECT * FROM books"
    cursor.execute(sql) # 5:Executing the query
    results = cursor.fetchall() # Fetching values from database
    lb=Label(win, text= "No. --Name--Author--publisher--year--edition", width=70)
    lbl.grid(row=6, columnspan=2)

    listbox = Listbox(win, width=70)
    listbox.grid(row=7, columnspan=2)
    for row in results: # Iterating and assigning values to list
        listbox.insert(END, row)
    conn.commit()# 6: Making the changes permanant in the database

win = Tk()
win.title("Book Details")

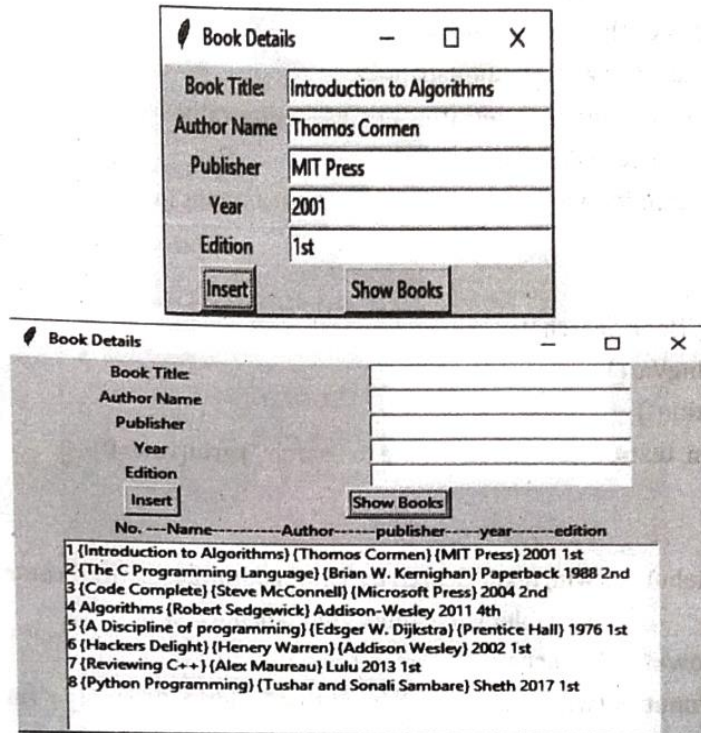
Label(win, text="Book Title:").grid(row=0)
Label(win, text="Author Name").grid(row=1)
Label(win, text="Publisher").grid(row=2)
Label(win, text="Year").grid(row=3)
Label(win, text="Edition").grid(row=4)

bTitle = Entry(win, width=30)
bTitle.grid(row=0, column=1)
aName = Entry(win,width=30)
aName.grid(row=1, column=1)
pub = Entry(win,width=30)
pub.grid(row=2, column=1)
year = Entry(win,width=30)
year.grid(row=3, column=1)
edi = Entry(win,width=30)
edi.grid(row=4, column=1)

b1=Button(win, text='Insert', command=insertCall)
b1.grid(row=5, column=0, padx=24)
b2=Button(win, text='Show Books', command=showBooks)
b2.grid(row=5, column=1, sticky=W, padx=38)

mainloop()
```

Output :



b) Design a database application to search the specified record from the database.

Explanation : In following program we have taken one entry box, five radio button in group and one button. On the click of button as per the radio button selected we are searching the attributes of the book. The results are displayed in the list. If no result found it displays the message box. We have used here MySQL **like** clause to retrieve the results.

Note : To run below example we have used the same database that we have created for example "10) A" above.

Solution :

#10B.py

```

from tkinter import *
from tkinter import messagebox
import mysql.connector as mysql

def showBooks():
    conn = mysql.connect(user='root', password='root', host='127.0.0.1')
    cursor = conn.cursor()
    cursor.execute("USE Library")
    sql = "SELECT * FROM books where "+var.get()+" like '"+stext.get()+"'"
    cursor.execute(sql)
    results = cursor.fetchall()
    if not results:
        messagebox.showinfo(title='Sorry', message="No Record Found!")
    else:

```



```

lbl = Label(win, text= "Search Results")
lbl.grid(row=7, columnspan=2)
listbox = Listbox(win, width=70)
listbox.grid(row=8, columnspan=2)
for row in results: # Iterating and assigning values to list
    listbox.insert(END, row)
conn.commit()# 6: Making the changes permanant in the database

win = Tk()
win.title("Book Search")
var = StringVar()
var.set("title")
Label(win, text="Select the Attribute to Search:").grid(row=0)

r1 = Radiobutton(win, text="Search by Title",width=20, justify="center",
                 variable=var, value="title",anchor="w")
r1.grid(row=1,columnspan=2)
r2 = Radiobutton(win, text="Search by Author",width=20, justify="center",
                 variable=var, value="author",anchor="w")
r2.grid(row=2,columnspan=2)
r3 = Radiobutton(win, text="Search by Pubication", width=20, justify="center",
                 variable=var, value="publisher",anchor="w")
r3.grid(row=3,columnspan=2)
r4 = Radiobutton(win, text="Search by Year", width=20, justify="center",
                 variable=var, value="year",anchor="w")
r4.grid(row=4,columnspan=2)

l1=Label(win, text="Enter the text string to Search:")
l1.grid(row=5,column=0)

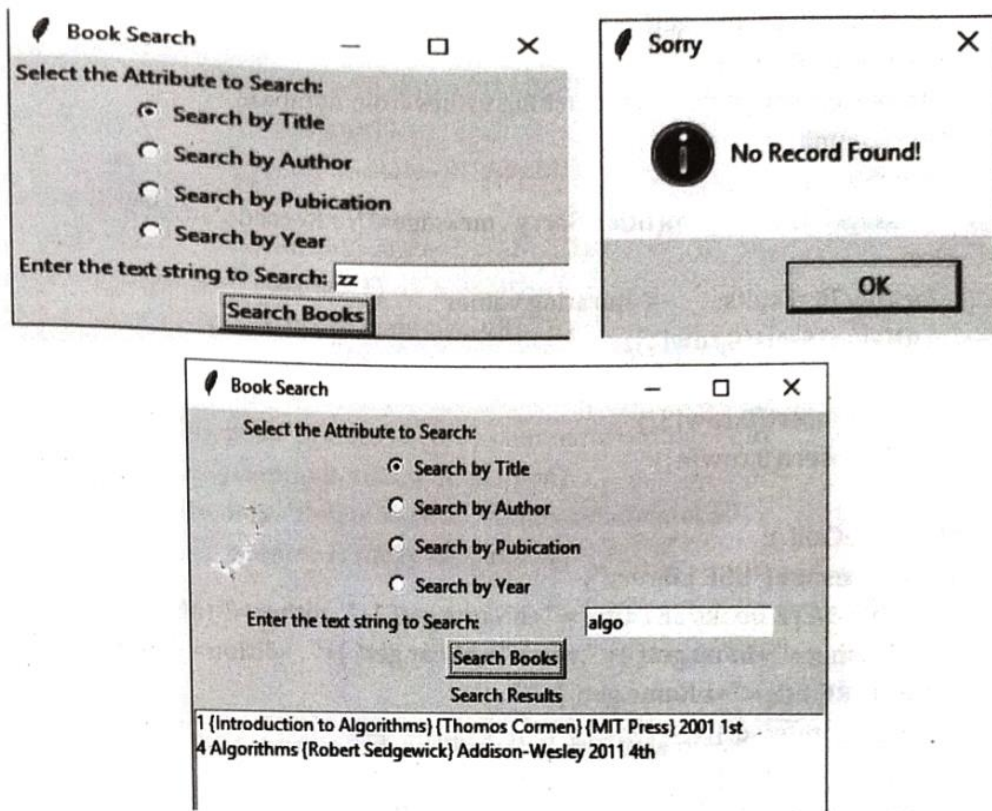
stext = Entry(win, width=20)
stext.grid(row=5,column=1)

b1=Button(win, text='Search Books', command=showBooks)
      .grid(row=6,columnspan=2)

mainloop()

```

Output :



c) Design a database application to that allows the user to add, delete and modify the records.

Explanation : In following program we used a text entry to first retrieve the record for update. Once we have displayed the records by using **callvalues()** method. After editing the information user can click Update Record button to make the changes in database by using **updateCall()** method. Also for deletion of a data entry user have to click Delete Record button by using **deleteCall()** method. If the particular record for updating or deleting is not found then user is notified with the message.

Solution :

#10C.py

```

from tkinter import *
from tkinter import messagebox
import mysql.connector as mysql

# Creating connection object and selecting current database to use.
conn = mysql.connect(user='root', password='root', host='127.0.0.1')
cursor = conn.cursor()

def callValues():
    bName.config(state=NORMAL) # Enabling text box
    aName.config(state=NORMAL) # Enabling text box
    bPub.config(state=NORMAL) # Enabling text box
    bYear.config(state=NORMAL) # Enabling text box
    bEdi.config(state=NORMAL) # Enabling text box
    cursor.execute("USE Library")
    
```



```

sql = "SELECT title,author,publisher,year,edition FROM books WHERE
      title='"+bName.get()+"'"
cursor.execute(sql)
results = cursor.fetchall() # Fetching values from database
conn.commit()
if not results:
    messagebox.showinfo(title='Sorry', message="No Record Found!")
else:
    for row in results: # Iterating values
        aName.insert(0,row[1])
        bPub.insert(0,row[2])
        bYear.insert(0,row[3])
        bEdi.insert(0,row[4])

def updateCall():
    cursor.execute("USE Library")
    sql = "UPDATE books SET title='"+bName.get()+"',author='"+aName.get()+"',
          publisher='"+bPub.get()+"',year="+ bYear.get()+" , edition=' "+bEdi.get()+"'
          WHERE title='"+bName.get()+"'"
    cursor.execute(sql)
    conn.commit()
    messagebox.showinfo(title='Confirmation', message="Information Updated")
    bName.delete(0,END) # Clearing text box
    aName.delete(0,END) # Clearing text box
    bPub.delete(0,END) # Clearing text box
    bYear.delete(0,END) # Clearing text box
    bEdi.delete(0,END) # Clearing text box

def deleteCall():
    cursor.execute("USE Library")
    sql = "DELETE FROM books where title='"+bName.get()+"'"
    cursor.execute(sql)
    conn.commit()
    messagebox.showinfo(title='Confirmation', message="Information Deleted")

win = Tk()
win.title("Book Updation Form")

Label(win, text="Enter Book Name").grid(row=0)
bName = Entry(win,width=35)
bName.grid(row=0, column=1)

b1=Button(win, text='Click here to get values', command=callValues)
b1.grid(row=1, sticky=N, pady=4,columnspan=2)

Label(win, text="Author").grid(row=2)
Label(win, text="Publication").grid(row=3)

```

```

Label(win, text="Year").grid(row=4)
Label(win, text="Edition").grid(row=5)

aName = Entry(win,width=35,state=DISABLED)
aName.grid(row=2, column=1)
bPub = Entry(win,width=35,state=DISABLED)
bPub.grid(row=3, column=1)
bYear = Entry(win,width=35,state=DISABLED)
bYear.grid(row=4, column=1)
bEdi = Entry(win,width=35,state=DISABLED)
bEdi.grid(row=5, column=1)

b2=Button(win, text='Update Record', command=updateCall)
b2.grid(row=6, column=0, sticky=N, pady=4)
b3=Button(win, text='Delete Record', command=deleteCall)
b3.grid(row=6, column=1, sticky=N, pady=4)

mainloop()
    
```

Output :

