#Basic Example

import sqlite3

#if needed to connect to Oracle database using cx\_Oracle

# import cx\_Oracle

#if needed to connect to MySql using MySQLDB

#import MySQLdb

connection = sqlite3.connect('employee.db') #connection object for creating connecition to the database

#If connection needs to be created for MySQLdb /Oracle DB

# connection = MySQLdb.connect(host ='localhost',database ='world',user = 'root',password ='nag123')

# connection = cx\_Oracle.connect('SYSTEM/nag123@localhost')

cursor = connection.cursor() #For executing the queries use cursor object

cursor.execute("drop table if exists emptab")

cursor.execute("create table emptab(empid int, empname char(20),salary int)")

cursor.execute("insert into emptab(empid , empname,salary) values(123,'Asha',10000)")

cursor.execute("insert into emptab(empid , empname,salary) values(123,'Dheeraj',10000)")

connection.commit()

cursor.execute("select \* from emptab")

# print(cursor.fetchall()) #to fetch all the records

# curosr.fetchone() #if one record needs to be fetched

# cursor.fetchmany(5) #number of records need to be provided as parameter.

cursor.execute("update emptab set salary = 20000 where empname ='Asha'")

connection.commit()

cursor.execute("select \*from emptab")

print(cursor.fetchall())

cursor.execute("delete from emptab where empname = 'Asha'")

connection.commit()

cursor.execute("select \* from emptab")

print(cursor.fetchall())

connection.close()

[(123, 'Asha', 20000), (123, 'Dheeraj', 10000)]

[(123, 'Dheeraj', 10000)]

*#writing the code in more pythonic way*

**import** sqlite3

**from** Employee **import** Employee

​

*#establish the connection with the database through sqlite3*

connection **=** sqlite3.connect("employee.db")

​

*#for testing purpose and if we want to have clean database at every time we run the code*

*# connection = sqlite3.connect(':memory:')*

​

​

​

*#Create a cursor object to execute the query on the database connection*

cursor **=** connection.cursor()

​

*#using the cursor work with database*

*#For the class create the objects now instead of just the values pass the class objects created*

​

emp1\_obj **=** Employee(234,"dheeraj",10000)

​

emp2\_obj **=** Employee(345,"Mary",20000)

​

emp3\_obj **=** Employee(567,"Lorry",30000)

​

*#one way of using insert statement to insert the values through pythonic way*

cursor.execute("insert into emptab(empid,empname,salary) values('{}','{}','{}')".

format(emp1\_obj.empid,emp1\_obj.empname,emp1\_obj.empsal))

​

​

*#second way of using insert statement*

*#get rid of format instead use tuples to pass the values , abd for place holder use ? , which means what ever specified in the*

*#values will be used in the place holder*

cursor.execute("insert into emptab(empid,empname,salary) values(?,?,?)",(emp2\_obj.empid,emp2\_obj.empname,emp2\_obj.empsal))

​

​

*#Third way of using insert statement and more readable instead of tuples use dictionary to pass values and the placeholder*

*#to be as named key*

cursor.execute("insert into emptab(empid,empname,salary) values(:empid,:empname,:empsal)",

{'empid':emp3\_obj.empid,'empname':emp3\_obj.empname,'empsal':emp3\_obj.empsal} 'empsal':emp3\_obj.empsal})

​

connection.commit()

​

cursor.execute("select \* from emptab where empname = 'dheeraj'")

print(cursor.fetchall())

​

cursor.execute("select \* from emptab where empname = ?",('Lorry',)) *#Since it is tuple and single need to give ,*

print(cursor.fetchall())

cursor.execute("select \* from emptab where empname = :empname",{'empname':'Mary'})

print(cursor.fetchall())

connection.close()

[(234, 'dheeraj', 10000), (234, 'dheeraj', 10000)]

[(567, 'Lorry', 30000), (567, 'Lorry', 30000), (567, 'Lorry', 30000), (567, 'Lorry', 30000), (567, 'Lorry', 30000), (567, 'Lorry', 30000)]

[(345, 'Mary', 20000), (345, 'Mary', 20000), (345, 'Mary', 20000), (345, 'Mary', 20000), (345, 'Mary', 20000), (345, 'Mary', 20000)]

**import** sqlite3

**from** Employee **import** Employee

​

*#establish the connection with the database through sqlite3*

connection **=** sqlite3.connect("employee.db")

*#for testing purpose and if we want to have clean database at every time we run the code*

*# connection = sqlite3.connect(':memory:')*

​

cursor **=** connection.cursor()

​

cursor.execute("drop table if exists EmployeeTable")

cursor.execute("create table EmployeeTable(empid int, empname char(20),salary int)")

​

emp1\_obj **=** Employee(234,"Harry",10000)

​

emp2\_obj **=** Employee(345,"Mary",20000)

​

emp3\_obj **=** Employee(567,"Lorry",30000)

​

**def** insert\_data(empobj):

**with** connection:

cursor.execute("insert into EmployeeTable(empid,empname,salary) values(:empid,:empname,:empsal)",

{'empid':empobj.empid,'empname':empobj.empname, 'empsal':empobj.empsal})

**def** update\_data\_pay(empobj,empsal):

**with** connection:

cursor.execute("update EmployeeTable set salary = :sal where empname = :empname",

{'empname':empobj.empname,'sal':empsal})

​

**def** select\_all():

cursor.execute("select \* from EmployeeTable")

print(cursor.fetchall())

​

​

**def** select\_emp\_name(empobj):

cursor.execute("select \* from EmployeeTable where empname=:name",{'name':empobj.empname})

print(cursor.fetchall())

​

**def** delete\_emp(empobj):

**with** connection:

cursor.execute("delete from EmployeeTable where empname=:name",{'name':empobj.empname})

*#call the functions to insert the data*

insert\_data(emp1\_obj)

insert\_data(emp2\_obj)

insert\_data(emp3\_obj)

​

*#call the function to display all*

select\_all()

​

*#call the function to update*

update\_data\_pay(emp1\_obj,40000)

​

select\_emp\_name(emp1\_obj)

​

delete\_emp(emp1\_obj)

​

select\_all()

​

​

​

​

[(234, 'Harry', 10000), (345, 'Mary', 20000), (567, 'Lorry', 30000)]

[(234, 'Harry', 40000)]

[(345, 'Mary', 20000), (567, 'Lorry', 30000)]

*#Still more pythonic way of wrting the code , instead of wrting insert statements multiple times*

**import** sqlite3

**from** tkinter **import** **\***

​

*#create root window*

root **=** Tk()

​

*#A function that takes employee number and displays the row*

​

**def** retrieve\_rows(empno):

*#connect to the database*

connection **=** sqlite3.connect('employee.db')

*#prepare the cursor*

cursor **=** connection.cursor()

cursor.execute("select \* from EmployeeTable where empid =:id",{'id':empno})

row **=** cursor.fetchone()

print(row)

*# if row is not None:*

lbl **=** Label(text **=** row , font**=**('Arial',14)).place(x**=**50,y**=**200)

cursor.close()

connection.close()

*#A function that takes input from the entry widget*

**def** display(self):

*#retrive the value from the entry widget*

str **=** e1.get()

*#display the values using label*

lbl **=** Label(text **=**'you entered:'**+**str,font **=**('Arial',14)).place(x**=**50,y**=**150)

*#call the function that retrieves the row*

retrieve\_rows(int(str))

*#create a frame as child to root window*

f **=** Frame(root,height**=**350,width **=**600)

​

*#let the frame not to shrik*

f.propagate(0)

​

*#attach the frame to root window*

f.pack()

​

*#create label*

l1 **=** Label(text **=**'Enter Employee Number:',font**=**('Arial',14))

​

*#create entry widget for accepting employee number*

e1 **=** Entry(f,width **=** 15, fg**=**'blue',bg**=**'yellow',font**=**('Arial',14))

*#when the user presses enter bind that event to display method*

e1.bind("<Return>",display)

​

*#place label and entry widgets in the frame*

l1.place(x**=**50,y**=**100)

e1.place(x**=**300,y**=**100)

​

*#handle the events*

root.mainloop()

​

​

None

​