# Introduction to Databases with Python and SQLite

### Aim

* Experience how data can be stored in a database.
* Be able to Create, and alter the structure of the database using DDL
* Be able to Add, Edit, and retrieve data using SQL
* Be able to visualise the structure of the data

## Tasks

You are going to create a database using SQLite3 and DDL (Data Definition Language)… We are going to use a SQLite db file. This is purely because this is easy to view what has happened in college (Using <https://inloop.github.io/sqlite-viewer/>) or at home using DB Browser for SQLite (from <https://sqlitebrowser.org/>). We expect you to use a DataBase Server (MySQL) for your computing project. The database you will create will hold some Student Information, Exam information and Exam results.

### Task1 – Create Form using Tkinter for Python

1. Create a new Forms program. For ease you can create a new button for each of the following tasks. So by the end you will have a form with lots of buttons on it…
2. Name the buttons something sensible (e.g. btnCreateDataBase).. You should also alter the display text so each button can be identified when the program runs.
3. At the start of the program file you need to add the following:

import sqlite3

from sqlite3 inport Error

### Task 2 - Create a Database

The code for creating the database is below… It will Create a new database file called Test.db, if does not already exist. It will create the file in the same folder as your Python code.

def createdatabase():

 try:

 sqlite3.connect("Test.db")

 messagebox.showinfo("Info","Database created successfully")

 except Error as e:

 messagebox.showinfo("Error","Database not created: "+str(e))

### Make your first button call the above procedure when it is clicked.Task 3 - Opening and Closing the connection to the Database

From now on before you execute the code for a following tasks the database connection needs to be opened… and after needs to be closed. I have created myconnection as global so it can be closed after this procedure has been run.

def opendatabase():

 global myconnection

 myconnection = None

 try:

 myconnection = sqlite3.connect("Test.db")

 messagebox.showinfo("Info","Database opened successfully")

 except Error as e:

 messagebox.showinfo("Error","Database not opened: "+str(e))

Ensure this command **myconnection.close()** is ALWAYS executed after connecting to the database

The Button command procedure will look like this

 def onclick():

 try:

 opendatabase()

 #insert future code between open and close

 myconnection.close()

 except Error as e:

 print(str(e))

### Task 4 – Creating tables

Now we are using DDL proper! We will be creating 3 tables: Students, Exams, Results I’ll give you the code for students and specify the other 2 tables…

def createstudenttable():

 DDLstr = """CREATE TABLE IF NOT EXISTS Students (

 Username text PRIMARY KEY,

 FirstName text,

 Surname text)"""

 try:

 opendatabase()

 mycommand = myconnection.cursor()

 mycommand.execute(DDLstr)

 myconnection.close()

 messagebox.showinfo("Info","Students Table created")

 except Error as e:

 messagebox.showinfo("Error","Students table not created: "+str(e))

Now create the Exams Table, which will hold information about the different exam each subject has

|  |  |
| --- | --- |
| Field | Properties |
| ExamCode | text , Primary Key |
| ExamTitle | Text |
| Subject | Text |

Now create the Results Table

|  |  |
| --- | --- |
| Field | Properties |
| ResultID | text , Primary Key |
| UserName | Text |
| ExamCode | Text |
| Score | Integer |

We will be adding extra fields to the 3 tables later

Open the sqlite database file and look at the tables that have been created… does everythong look correct?

### Task 5 Adding Fields to a table

Here is an example that adds a field to an existing table:

def createfield():

 DDLstr = """ALTER TABLE Exams ADD COLUMN Weight real"""

 try:

 opendatabase()

 mycommand = myconnection.cursor()

 mycommand.execute(DDLstr)

 myconnection.close()

 messagebox.showinfo("Info","Weight column added")

 except Error as e:

 messagebox.showinfo("Error","Weight column not added: "+str(e))

Now add the following:

 'DDLstr = "ALTER TABLE Results ADD COLUMN RawScore real"

 'Add MaxScore field to the Exams Table with type integer

 'Add a DateofBirth Field to the Students Table

 'Add a Gender Field to Students Table

 'Add ExamDate to Results (just storing the year as an integer)

### Task 6 Adding data to a table

Let’s start with adding some hard coded data (you will never do this in practice but that way you can see the syntax…

def insertonerecord():

 SQLstr = """INSERT INTO Students(Username, FirstName, Surname) VALUES ('BB','Robert','theBuilder')"""

 try:

 opendatabase()

 mycommand = myconnection.cursor()

 mycommand.execute(SQLstr)

 myconnection.commit()

 myconnection.close()

 messagebox.showinfo("Info","Record inserted")

 except Error as e:

 messagebox.showinfo("Error","Record not inserted: "+str(e))

Make sure you understand the syntax…

* Why are sections in single quotes?

Now we will do the same thing again BUT with data stored in variables

def insertanotherrecord():

 uName = "WB"

 fName = "Wendy"

 sName = "theBuilder"

 SQLstr = f"""INSERT INTO Students(Username, FirstName, Surname) VALUES ('{uName}','{fName}','{sName}')"""

 try:

 opendatabase()

 mycommand = myconnection.cursor()

 mycommand.execute(SQLstr)

 myconnection.commit()

 myconnection.close()

 messagebox.showinfo("Info","Record inserted")

 except Error as e:

 messagebox.showinfo("Error","Record not inserted: "+str(e))

Alter the Procedure to add the following data to the **EXAMS** Table. The column headings are the exact field names..

| **ExamCode** | **ExamTitle** | **Subject** | **Weight** | **MaxScore** |
| --- | --- | --- | --- | --- |
| COMP999 | Geekdom | Computing | 0.6 | 100 |

Add Textboxes to your MainForm to allow you to enter data into the **Students** and **Exams** Tables.
(probably best to handle as separate events )

### Task 7 Adding data to the database from CSV files.

It is often useful to have sets of test data stored as text files. As it happens most DBMS also have CSV import… but that would be no fun! It gives us a chance to see a sound way to manipulate files in .net.

Here is an example for the Student database… Make sure this inserts correctly

**I haven’t converted the rest from VB to Python yet, so have a go at this yourself…**

Sub InserttoStudentFromCsv()

' here is an insertion of a set of records taken from a correctly formatted CSV file.

'You need to have put the CSV files into the Debug bin folder

‘get them here <https://drive.google.com/file/d/0B1nECgSBg0g7S0NDamxOTDFWM0E/view?usp=sharing>

Dim SQLstr As String

Dim SR As New StreamReader("students.csv") 'hope you have this file in the debug bin folder!

Dim txtLine As String

Dim FieldStr As String()

 Do Until SR.EndOfStream ' keeps going until end of file

 Try

 txtLine = SR.ReadLine()

 FieldStr = txtLine.Split(",")

 SQLstr = "INSERT INTO Students([Username], [FirstName], [Surname]) VALUES ('" & FieldStr(0) & "','" & FieldStr(1) & "','" & FieldStr(2) & "')"

 mycommand = New OleDbCommand(SQLstr, myconnection)

 mycommand.ExecuteNonQuery()

 Console.WriteLine("inserted")

 Catch ex As Exception

 Console.WriteLine(ex.Message)

 End Try

 Loop

End Sub

 Alter the above code to insert the csv data for the Exams, Users and Results Tables…

Sample Python Solution

from tkinter import \*

from tkinter import messagebox

import sqlite3

from sqlite3 import Error

win = Tk()

win.geometry("1200x600")

username\_var=StringVar()

firstname\_var=StringVar()

surname\_var=StringVar()

dob\_var=StringVar()

gender\_var=StringVar()

def createdatabase():

 try:

 sqlite3.connect("Test.db")

 messagebox.showinfo("Info","Database created successfully")

 except Error as e:

 messagebox.showinfo("Error","Database not created: "+str(e))

def opendatabase():

 global myconnection

 myconnection = None

 try:

 myconnection = sqlite3.connect("Test.db")

 messagebox.showinfo("Info","Database opened successfully")

 except Error as e:

 messagebox.showinfo("Error","Database not opened: "+str(e))

def closedatabase():

 global myconnection

 try:

 myconnection.close()

 messagebox.showinfo("Info","Database closed successfully")

 except Error as e:

 messagebox.showinfo("Error","Database not closed: "+str(e))

def onclick():

 opendatabase()

 #insert future code between open and close

 myconnection.close()

def createstudenttable():

 DDLstr = """CREATE TABLE IF NOT EXISTS Students (

 Username text PRIMARY KEY,

 FirstName text,

 Surname text)"""

 try:

 opendatabase()

 mycommand = myconnection.cursor()

 mycommand.execute(DDLstr)

 myconnection.close()

 messagebox.showinfo("Info","Students Table created")

 except Error as e:

 messagebox.showinfo("Error","Students table not created: "+str(e))

def createexamstable():

 DDLstr = """CREATE TABLE IF NOT EXISTS Exams (

 ExamCode text PRIMARY KEY,

 ExamTitle text,

 Subject text)"""

 try:

 opendatabase()

 mycommand = myconnection.cursor()

 mycommand.execute(DDLstr)

 myconnection.close()

 messagebox.showinfo("Info","Exams Table created")

 except Error as e:

 messagebox.showinfo("Error","Exams table not created: "+str(e))

def createresultstable():

 DDLstr = """CREATE TABLE IF NOT EXISTS Results (

 ResultID text PRIMARY KEY,

 UserName text,

 ExamCode text,

 Score integer)"""

 try:

 opendatabase()

 mycommand = myconnection.cursor()

 mycommand.execute(DDLstr)

 myconnection.close()

 messagebox.showinfo("Info","Results Table created")

 except Error as e:

 messagebox.showinfo("Error","Results table not created: "+str(e))

def createfield():

 DDLstr = """ALTER TABLE Exams ADD COLUMN Weight real"""

 try:

 opendatabase()

 mycommand = myconnection.cursor()

 mycommand.execute(DDLstr)

 myconnection.close()

 messagebox.showinfo("Info","Weight column added")

 except Error as e:

 messagebox.showinfo("Error","Weight column not added: "+str(e))

def runSQL(sql):

 try:

 opendatabase()

 mycommand = myconnection.cursor()

 mycommand.execute(sql)

 myconnection.close()

 messagebox.showinfo("Info","SQL executed successfully")

 except Error as e:

 messagebox.showinfo("Error","SQL '" +sql+ "' failed: "+str(e))

def createextrafields():

 runSQL("""ALTER TABLE Results ADD COLUMN RawScore real""")

 runSQL("""ALTER TABLE Exams ADD COLUMN MaxScore integer""")

 runSQL("""ALTER TABLE Students ADD COLUMN DateOfBirth text""")

 runSQL("""ALTER TABLE Students ADD COLUMN Gender text""")

 runSQL("""ALTER TABLE Results ADD COLUMN ExamDate integer""")

def insertonerecord():

 SQLstr = """INSERT INTO Students(Username, FirstName, Surname)

 VALUES ('BB','Robert','theBuilder')"""

 try:

 opendatabase()

 mycommand = myconnection.cursor()

 mycommand.execute(SQLstr)

 myconnection.commit()

 myconnection.close()

 messagebox.showinfo("Info","Record inserted")

 except Error as e:

 messagebox.showinfo("Error","Record not inserted: "+str(e))

def insertanotherstudentrecord():

 uName = "WB"

 fName = "Wendy"

 sName = "theBuilder"

 SQLstr = f"""INSERT INTO Students(Username, FirstName, Surname) VALUES ('{uName}','{fName}','{sName}')"""

 try:

 opendatabase()

 mycommand = myconnection.cursor()

 mycommand.execute(SQLstr)

 myconnection.commit()

 myconnection.close()

 messagebox.showinfo("Info","Record inserted")

 except Error as e:

 messagebox.showinfo("Error","Record not inserted: "+str(e))

def insertexamrecord():

 SQLstr = """INSERT INTO Exams(ExamCode, ExamTitle, Subject, Weight, MaxScore)

 VALUES ('COMP999','Geekdom','Computing', 0.6, 100)"""

 try:

 opendatabase()

 mycommand = myconnection.cursor()

 mycommand.execute(SQLstr)

 myconnection.commit()

 myconnection.close()

 messagebox.showinfo("Info","Exam record inserted")

 except Error as e:

 messagebox.showinfo("Error","Exam record not inserted: "+str(e))

def addstudent():

 bIns = False

 SQLstr = f"""INSERT INTO Students(Username, FirstName, Surname, DateOfBirth, Gender)

 VALUES ('{username\_var.get()}','{firstname\_var.get()}','{surname\_var.get()}','{dob\_var.get()}','{gender\_var.get()}')"""

 try:

 opendatabase()

 mycommand = myconnection.cursor()

 mycommand.execute(SQLstr)

 myconnection.commit()

 myconnection.close()

 messagebox.showinfo("Info","Student record inserted")

 bIns = True

 username\_var.set("")

 firstname\_var.set("")

 surname\_var.set("")

 dob\_var.set("")

 gender\_var.set("")

 except Error as e:

 if bIns:

 messagebox.showinfo("Error","Student Record not inserted: "+str(e))

 else:

 messagebox.showinfo("Error","Failed to reset fields: "+str(e))

Button(win, text="Create Database", height=5, width=25, command=lambda:createdatabase()).grid(row=0,column=0)

Button(win, text="Open Database", height=5, width=25, command=lambda:onclick()).grid(row=0,column=1)

Button(win, text="Create Students table", height=5, width=25, command=lambda:createstudenttable()).grid(row=0,column=2)

Button(win, text="Create Exams table", height=5, width=25, command=lambda:createexamstable()).grid(row=0,column=3)

Button(win, text="Create Results table", height=5, width=25, command=lambda:createresultstable()).grid(row=0,column=4)

Button(win, text="Create Field", height=5, width=25, command=lambda:createfield()).grid(row=1,column=0)

Button(win, text="Create Extra Fields", height=5, width=25, command=lambda:createextrafields()).grid(row=1,column=1)

Button(win, text="Insert record", height=5, width=25, command=lambda:insertonerecord()).grid(row=1,column=2)

Button(win, text="Insert Another Record", height=5, width=25, command=lambda:insertanotherstudentrecord()).grid(row=1,column=3)

Button(win, text="Insert An Exam Record", height=5, width=25, command=lambda:insertexamrecord()).grid(row=1,column=4)

Label(win, text="Username").grid(row=2,column=0)

Label(win, text="FirstName").grid(row=2,column=1)

Label(win, text="Surname").grid(row=2,column=2)

Label(win, text="DateOfBirth").grid(row=2,column=3)

Label(win, text="Gender").grid(row=2,column=4)

Entry(win, textvariable = username\_var).grid(row=3, column=0)

Entry(win, textvariable = firstname\_var).grid(row=3, column=1)

Entry(win, textvariable = surname\_var).grid(row=3, column=2)

Entry(win, textvariable = dob\_var).grid(row=3, column=3)

Entry(win, textvariable = gender\_var).grid(row=3, column=4)

Button(win, text="Insert Student Details", height=5, width=25, command=lambda:addstudent()).grid(row=2,column=5)

win.mainloop()