Project topic: Database Application SQlite3

1. Background

SQLite is a serverless, embedded relational database, in-process library that implements a self-contained, that is, it doesn't require a dedicated management database system to use, as the database is literarily part of your code and is not an outside resource, hence; no required software as it is part of standard library. It requires zero-configuration, and transactional SQL database engine. The code for SQLite is in the public domain and is thus free for use for any purpose, commercial or private. Furthermore, it also can be used for testing and prototyping out an application. It's simple to use, and database can be a single file, as it can be also memory database. In addition, SQLite3 database application is easily updated.

2. Structure of the solution

We have structured the solution in different parts. They are listed below:

1) Importing windows from the Tkinter module
2) Define CRUD operations
3) Make the widget for GUI-like buttons, text entry, labels, tree view

3. Detailed description of the solution

First we import libraries, thereafter we imported all from Tkinter which is the library for Gui. Thereafter we imported sqLite3, in other to connect to Database. GUI widgets were created by importing ttk another module as ttk. The message box was created by importing tkMessagesBox module as tkMessageBox.

For the position and structure of the main page we have used window object from the tkinter library and then divided the window into frames. Frames are

further divided into grids so we can place our button, label, text entry, tree view widgets. All of these widgets are from tkinter libraries.

Then we have used subroutines to communicate and pass data to the database from by utilizing the different options provided by tkinter gui for handling simple call backs. The CRUD subroutines are the backbone of this application and they are mentioned below:

3.1 Detailed description of subprograms

In other to create and connect different command of the database line, the following subprograms where defined.

Define crud functions were used to create and connect different command of the database line.

def Database():

This subprogram was set as a function to open the database connection. In here the table is created if it does exist or it will connect if the table already exists. This is the most important function while working with databases in Sqlite as it opens the database connection for crud operations.

def Create():

The def create subprogram is a function used for entering new data to the table created in the database. In our implementation we have used if conditions to validate if the data fields are not empty and if it is all validated then it will call the database to create a new row and add it the table with respective value. If the field are empty we have configured a text widget that will show the datafield in the entry boxes or widgets are empty. After this function is executed an auto-incremented unique id will be generated so we can later distinguish each data separately.

Def update ():

This subprogram updates value if they are empty and also update them.

This function helps to update the existing database if some entries have been wrong and needs to be re-entered. We have implemented such that the ID from the database needs to be given and then any entry widget data will be updated to the row with respective ID number. Just for validating that the user is not populating data with empty values we have used if condition to validate and then text widet to give the message if the entry is added or not.

def Read():

The purpose of this function is to read all data that has been inputted. It will list all the datas in the database table and pass it to the tree-view widget from tkinter and then display all the data in the table with respective rows.

def Exit(): The exit function, just has the name implies ends the program, when the user inputs a yes.