КОД ПРОГРАММЫ:

```
import tkinter as tk
from tkinter import ttk
import sqlite3
from tkinter import messagebox


class DatabaseApp:
    def __init__(self, master, connection_params):
        self.master = master
        self.connection_params  = connection_params
        self.master.title("lab 5")
        self.setup_styles()
        self.notebook = ttk.Notebook(master)
        self.notebook.pack(expand=True, fill='both')
        self.conn = sqlite3.connect(**connection_params)
        self.cursor = self.conn.cursor()
        self.table_names = self.get_table_names()
        for table_name in self.table_names:
            frame = tk.Frame(self.notebook)
            self.notebook.add(frame, text=table_name)
            self.create_table_view(frame, table_name)

    def setup_styles(self):
        style = ttk.Style()
        style.configure("TNotebook", tabposition='n')
        style.configure("TButton", padding=6, relief="flat", background="#4CAF50", foreground="black")
        style.map("TButton", background=[("active", "#45a049")], foreground=[("active", "black")])
        style.configure("Treeview", rowheight=25, font=('Arial', 10))
        style.configure("Treeview.Heading", font=('Arial', 12, 'bold'))

    def get_table_names(self):
        # Fetch table names from the database
        self.cursor.execute("SELECT name FROM sqlite_master WHERE type='table';")
        table_names = [row[0] for row in self.cursor.fetchall()]
        return table_names

    def create_table_view(self, frame, table_name):
        # Fetch column names
        self.cursor.execute(f"PRAGMA table_info({table_name});")
        columns = [row[1] for row in self.cursor.fetchall()]
        tree = ttk.Treeview(frame, columns=columns, show='headings', selectmode='browse')
        tree.pack(expand=True, fill='both')
        for col in columns:
            tree.heading(col, text=col)
            tree.column(col, width=100, anchor='center')
        self.populate_treeview(tree, table_name)
        button_frame = tk.Frame(frame)
        button_frame.pack(pady=10)
        add_button = ttk.Button(button_frame, text="Добавить", command=lambda: self.add_row(tree, table_name))
        add_button.pack(side=tk.LEFT, padx=10)
        delete_button = ttk.Button(button_frame, text="Удалить", command=lambda: self.delete_row(tree, table_name))
        delete_button.pack(side=tk.LEFT, padx=10)
        edit_button = ttk.Button(button_frame, text="Изменить", command=lambda: self.edit_row(tree, table_name))
        edit_button.pack(side=tk.LEFT, padx=10)
        refresh_button = ttk.Button(button_frame, text="Обновить", command=lambda: self.populate_treeview(tree, 
table_name))
        refresh_button.pack(side=tk.LEFT, padx=10)

    def populate_treeview(self, tree, table_name):
        self.cursor.execute(f"SELECT * FROM {table_name};")
        data = self.cursor.fetchall()
        tree.delete(*tree.get_children())
        for row in data:
            tree.insert('', 'end', values=row)

    def add_row(self, tree, table_name):
        self.cursor.execute(f"PRAGMA table_info({table_name});")
        columns = [row[1] for row in self.cursor.fetchall()]
        add_dialog = tk.Toplevel(self.master)
        add_dialog.title("Добавить строку")
        entry_widgets = []
        for col in columns:
            label = tk.Label(add_dialog, text=col)
            label.grid(row=columns.index(col), column=0, padx=10, pady=5, sticky='e')
            entry = tk.Entry(add_dialog)
            entry.grid(row=columns.index(col), column=1, padx=10, pady=5, sticky='w')
            entry_widgets.append(entry)

        def insert_row():
            values = [entry.get() for entry in entry_widgets]
            placeholders = ', '.join(['?' for _ in values])
            query = f"INSERT INTO {table_name} VALUES ({placeholders});"
            self.cursor.execute(query, values)
            self.conn.commit()
            self.populate_treeview(tree, table_name)
            add_dialog.destroy()

        submit_button = ttk.Button(add_dialog, text="Подтвердить", command=insert_row)
        submit_button.grid(row=len(columns), columnspan=2, pady=10)

    def delete_row(self, tree, table_name):
        selected_item = tree.selection()
        if not selected_item:
            messagebox.showwarning("Предупреждение", "Пожалуйста, выберите строку для удаления.")
            return
        confirm = messagebox.askyesno("Подтверждение", "Вы уверены, что хотите удалить эту строку?")
        if not confirm:
            return
        values = tree.item(selected_item)['values']
        where_clause = ' AND '.join([f"{column} = ?" for column in tree['columns']])
        query = f"DELETE FROM {table_name} WHERE {where_clause};"
        self.cursor.execute(query, values)
        self.conn.commit()
        self.populate_treeview(tree, table_name)

    def edit_row(self, tree, table_name):
        selected_item = tree.selection()
        if not selected_item:
            messagebox.showwarning("Предупреждение", "Пожалуйста, выберите строку для изменения.")
            return
        values = tree.item(selected_item)['values']
        self.cursor.execute(f"PRAGMA table_info({table_name});")
        columns = [row[1] for row in self.cursor.fetchall()]
        edit_dialog = tk.Toplevel(self.master)
        edit_dialog.title("Изменить строку")
        entry_widgets = []
        for col, value in zip(columns, values):
            label = tk.Label(edit_dialog, text=col)
            label.grid(row=columns.index(col), column=0, padx=10, pady=5, sticky='e')
            entry = tk.Entry(edit_dialog)
            entry.insert(0, value)
            entry.grid(row=columns.index(col), column=1, padx=10, pady=5, sticky='w')
            entry_widgets.append(entry)

        def update_row():
            new_values = [entry.get() for entry in entry_widgets]
            set_clause = ', '.join([f"{column} = ?" for column in columns])
            where_clause = ' AND '.join([f"{column} = ?" for column in columns])
            query = f"UPDATE {table_name} SET {set_clause} WHERE {where_clause};"
            self.cursor.execute(query, new_values + values)
            self.conn.commit()
            self.populate_treeview(tree, table_name)
            edit_dialog.destroy()

        submit_button = ttk.Button(edit_dialog, text="Подтвердить", command=update_row)
        submit_button.grid(row=len(columns), columnspan=2, pady=10)
```