```
#encoding:utf-8

import csv
from tkinter import *
from tkinter import messagebox
import sqlite3

def extraer_datos(fichero):
    try:
        with open(fichero) as f:
            l = [row for row in csv.reader(f, delimiter=';', quotechar='"')]
        return l[1:] # elimina la linea de la cabecera
    except:
        messagebox.showerror("Error", "Error en la apertura del fichero de libros" )
        return None

def almacenar_bd(libros):
    conn = sqlite3.connect('books.db')
    conn.text_factory = str  # para evitar problemas con el conjunto de caracteres que maneja la BD
    conn.execute("DROP TABLE IF EXISTS BOOKS")
    conn.execute('''CREATE TABLE BOOKS
      (ISBN        CHAR(9) PRIMARY KEY,
      TITLE         TEXT    NOT NULL,
      AUTHOR        TEXT    NOT NULL,
      YEAR         INTEGER    NOT NULL,
      PUBLISHER       TEXT NOT NULL);''')

    for i in libros:
        if i[3] == 'Unknown':
            i[3] = 0
        conn.execute("""INSERT INTO BOOKS (ISBN, TITLE, AUTHOR, YEAR, PUBLISHER) VALUES
(?,?,?,?,?)""",(i[0],i[1],i[2],int(i[3]),i[4]))
    conn.commit()
    cursor = conn.execute("SELECT COUNT(*) FROM BOOKS")
    messagebox.showinfo( "Base Datos", "Base de datos creada correctamente \nHay " +
str(cursor.fetchone()[0]) + " registros")
    conn.close()

def cargar():
    respuesta = messagebox.askyesno(title="Confirmar",message="Esta seguro que quiere recargar
los datos?")
    if respuesta:
        libros = extraer_datos("books.csv")
        if libros:
            almacenar_bd(libros)

def listar(cursor):
    v = Toplevel()
```

```
    sc = Scrollbar(v)
    sc.pack(side=RIGHT, fill=Y)
    lb = Listbox(v, width=150, yscrollcommand=sc.set)
    for row in cursor:
        s = 'TITULO: ' + row[1]
        lb.insert(END, s)
        lb.insert(END, "--------------------------------------------------------------------------")
        s = "     ISBN: " + row[0] + ' | AUTOR: ' + row[2]+ ' | AÑO: ' + (str(row[3]) if row[3] != 0 else
"Desconocido")
        lb.insert(END, s)
        lb.insert(END,"\n\n")
    lb.pack(side=LEFT, fill=BOTH)
    sc.config(command=lb.yview)

def listar_editorial(cursor):
    v = Toplevel()
    sc = Scrollbar(v)
    sc.pack(side=RIGHT, fill=Y)
    lb = Listbox(v, width=150, yscrollcommand=sc.set)
    for row in cursor:
        s = 'TITULO: ' + row[0]
        lb.insert(END, s)
        lb.insert(END, "--------------------------------------------------------------------------")
        s = '     AUTOR: ' + row[1] + ' | EDITORIAL: ' + row[2]
        lb.insert(END, s)
        lb.insert(END,"\n\n")
    lb.pack(side=LEFT, fill=BOTH)
    sc.config(command=lb.yview)

def listar_completo():
    conn = sqlite3.connect('books.db')
    conn.text_factory = str
    cursor = conn.execute("SELECT ISBN, TITLE, AUTHOR, YEAR FROM BOOKS")
    conn.close
    listar(cursor)


def listar_ordenado():
    def lista():
        conn = sqlite3.connect('books.db')
        conn.text_factory = str
        if control.get() == 1:
            cursor = conn.execute("SELECT ISBN, TITLE, AUTHOR, YEAR FROM BOOKS ORDER BY
ISBN")
        else:
            cursor = conn.execute("SELECT ISBN, TITLE, AUTHOR, YEAR FROM BOOKS ORDER BY
YEAR")
        conn.close
```

```
            listar(cursor)
    ventana = Toplevel()
    control = IntVar()
    rb1 = Radiobutton(ventana, text="Ordenado por Año", variable=control, value=0)
    rb2 = Radiobutton(ventana, text="Ordenado por ISBN", variable=control, value=1)
    b = Button(ventana, text="Listar", command=lista)
    rb1.pack()
    rb2.pack()
    b.pack()

def buscar_editorial():
    def lista(event):
            conn = sqlite3.connect('books.db')
            conn.text_factory = str
            cursor = conn.execute("SELECT TITLE, AUTHOR, PUBLISHER FROM BOOKS WHERE
PUBLISHER = '" + sb.get() +"'")
            conn.close
            listar_editorial(cursor)

    conn = sqlite3.connect('books.db')
    conn.text_factory = str
    cursor = conn.execute("SELECT DISTINCT PUBLISHER FROM BOOKS")

    editoriales = [i[0] for i in cursor]

    v = Toplevel()
    sb = Spinbox(v, values=editoriales)
    sb.bind("<Return>", lista)
    sb.pack()

    conn.close()

def buscar_titulo():
    def lista(event):
            conn = sqlite3.connect('books.db')
            conn.text_factory = str
            cursor = conn.execute("SELECT ISBN, TITLE, AUTHOR, YEAR FROM BOOKS WHERE TITLE LIKE
'%" + en.get() +"%'")
            conn.close
            listar(cursor)

    conn = sqlite3.connect('books.db')
    conn.text_factory = str

    v = Toplevel()
    lb = Label(v, text="Introduzca la palabra a buscar")
    en = Entry(v)
    en.bind("<Return>", lista)
```

```
    lb.pack(side=LEFT)
    en.pack(side=LEFT)

    conn.close()

def ventana_principal():
    raiz = Tk()

    menu = Menu(raiz)

    #DATOS
    menudatos = Menu(menu, tearoff=0)
    menudatos.add_command(label="Cargar", command=cargar)
    menudatos.add_command(label="Salir", command=raiz.quit)
    menu.add_cascade(label="Datos", menu=menudatos)

    #LISTAR
    menulistar = Menu(menu, tearoff=0)
    menulistar.add_command(label="Completo", command=listar_completo)
    menulistar.add_command(label="Ordenado", command=listar_ordenado)
    menu.add_cascade(label="Listar", menu=menulistar)

    #BUSCAR
    menubuscar = Menu(menu, tearoff=0)
    menubuscar.add_command(label="Título", command=buscar_titulo)
    menubuscar.add_command(label="Editorial", command=buscar_editorial)
    menu.add_cascade(label="Buscar", menu=menubuscar)

    raiz.config(menu=menu)

    raiz.mainloop()



if __name__ == "__main__":
    ventana_principal()
```