

```

import sqlite3
import tkinter as tk
from tkinter import ttk, messagebox
from datetime import datetime

class BusinessApp:
    def __init__(self, root):
        self.root = root
        self.root.title("S-CRM: Управління Складом 2026")
        self.root.geometry("900x600")

        self.db_init()
        self.create_widgets()
        self.update_tables()

    def db_init(self):
        self.conn = sqlite3.connect("business.db")
        self.cur = self.conn.cursor()
        self.cur.execute("""CREATE TABLE IF NOT EXISTS inventory (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            name TEXT, quantity INTEGER, price REAL)""")
        self.cur.execute("""CREATE TABLE IF NOT EXISTS sales (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            name TEXT, quantity INTEGER, total REAL, date TEXT)""")
        self.conn.commit()

    def create_widgets(self):
        self.tabs = ttk.Notebook(self.root)
        self.tab_inv = ttk.Frame(self.tabs)
        self.tab_sales = ttk.Frame(self.tabs)
        self.tabs.add(self.tab_inv, text="📦 Склад ")
        self.tabs.add(self.tab_sales, text="💰 Продажі ")
        self.tabs.pack(expand=1, fill="both")

        # --- Склад ---
        f_left = tk.LabelFrame(self.tab_inv, text="Додати товар", padx=10, pady=10)
        f_left.pack(side="left", fill="y", padx=10, pady=10)

        tk.Label(f_left, text="Назва:").pack()
        self.ent_name = tk.Entry(f_left)
        self.ent_name.pack(pady=5)

        tk.Label(f_left, text="Кількість:").pack()
        self.ent_qty = tk.Entry(f_left)
        self.ent_qty.pack(pady=5)

        tk.Label(f_left, text="Ціна (за од.):").pack()
        self.ent_price = tk.Entry(f_left)

```

```

self.ent_price.pack(pady=5)

tk.Button(f_left, text="Зберегти на склад", bg="#d4edda",
command=self.add_to_inv).pack(fill="x", pady=10)
tk.Button(f_left, text="Видалити обране", bg="#f8d7da",
command=self.delete_inv).pack(fill="x")

self.tree_inv = ttk.Treeview(self.tab_inv, columns=("ID", "Назва", "К-сть", "Ціна"),
show="headings")
for col in ("ID", "Назва", "К-сть", "Ціна"): self.tree_inv.heading(col, text=col)
self.tree_inv.pack(side="right", expand=1, fill="both", padx=10, pady=10)

# --- Продажі ---
f_top = tk.Frame(self.tab_sales, pady=10)
f_top.pack(fill="x")

tk.Label(f_top, text="Кількість для продажу:").pack(side="left", padx=10)
self.ent_sell_qty = tk.Entry(f_top, width=10)
self.ent_sell_qty.pack(side="left")

tk.Button(f_top, text="Оформити продаж", bg="#cfe2ff",
command=self.make_sale).pack(side="left", padx=10)

self.tree_sales = ttk.Treeview(self.tab_sales, columns=("ID", "Товар", "К-сть", "Сума",
"Дата"), show="headings")
for col in ("ID", "Товар", "К-сть", "Сума", "Дата"): self.tree_sales.heading(col, text=col)
self.tree_sales.pack(expand=1, fill="both", padx=10, pady=10)

def add_to_inv(self):
    try:
        name, qty, prc = self.ent_name.get(), int(self.ent_qty.get()), float(self.ent_price.get())
        self.cur.execute("INSERT INTO inventory (name, quantity, price) VALUES (?, ?, ?)",
(name, qty, prc))
        self.conn.commit()
        self.update_tables()
        for e in [self.ent_name, self.ent_qty, self.ent_price]: e.delete(0, 'end')
    except: messagebox.showerror("Error", "Некоректні дані")

def make_sale(self):
    selected = self.tree_inv.selection()
    if not selected: return messagebox.showwarning("!", "Оберіть товар на складі")

    item_id = self.tree_inv.item(selected[0])['values'][0]
    curr_name = self.tree_inv.item(selected[0])['values'][1]
    curr_qty = self.tree_inv.item(selected[0])['values'][2]
    price = self.tree_inv.item(selected[0])['values'][3]

    try:

```

```

sell_qty = int(self.ent_sell_qty.get())
if sell_qty > curr_qty: raise ValueError("Немає стільки на складі")

total = sell_qty * price
date = datetime.now().strftime("%d.%m.%Y %H:%M")

self.cur.execute("UPDATE inventory SET quantity = quantity - ? WHERE id = ?",
(sell_qty, item_id))
self.cur.execute("INSERT INTO sales (name, quantity, total, date) VALUES (?, ?, ?,
?)", (curr_name, sell_qty, total, date))
self.conn.commit()
self.update_tables()
messagebox.showinfo("Успіх", f"Продано на суму: {total} грн")
except Exception as e: messagebox.showerror("Error", str(e))

def delete_inv(self):
    selected = self.tree_inv.selection()
    if selected:
        self.cur.execute("DELETE FROM inventory WHERE id = ?",
(self.tree_inv.item(selected[0])['values'][0],))
        self.conn.commit()
        self.update_tables()

def update_tables(self):
    for t in [self.tree_inv, self.tree_sales]:
        for row in t.get_children(): t.delete(row)

self.cur.execute("SELECT * FROM inventory")
for r in self.cur.fetchall(): self.tree_inv.insert("", "end", values=r)

self.cur.execute("SELECT * FROM sales")
for r in self.cur.fetchall(): self.tree_sales.insert("", "end", values=r)

if __name__ == "__main__":
    root = tk.Tk()
    app = BusinessApp(root)
    root.mainloop()

```