02.15 7:18 PM

```
import sqlite3
import tkinter as tk
from tkinter import ttk, messagebox
import matplotlib.pyplot as plt

# Database Setup
conn = sqlite3.connect("expenses.db")
cursor = conn.cursor()
cursor.execute('''
    CREATE TABLE IF NOT EXISTS expenses (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        amount REAL,
        category TEXT,
        description TEXT,
        date TEXT
    )
''')
conn.commit()

# Function to add an expense
def add_expense():
    amount = amount_entry.get()
    category = category_combobox.get()
    description = description_entry.get()
    date = date_entry.get()

    if not amount or not category or not date:
        messagebox.showerror("Error", "Please fill all required fields!")
        return

    try:
        amount = float(amount)
        cursor.execute("INSERT INTO expenses (amount, category, description, date) VALUES
(?, ?, ?, ?)",
                       (amount, category, description, date))
        conn.commit()
        messagebox.showinfo("Success", "Expense added successfully!")
        load_expenses()
    except ValueError:
        messagebox.showerror("Error", "Invalid amount!")

# Function to load expenses into the table
def load_expenses():
    for row in tree.get_children():
        tree.delete(row)

    cursor.execute("SELECT * FROM expenses")
    for row in cursor.fetchall():
        tree.insert("", tk.END, values=row)

# Function to analyze expenses
def analyze_expenses():
    cursor.execute("SELECT category, SUM(amount) FROM expenses GROUP BY category")
    data = cursor.fetchall()

    if not data:
        messagebox.showinfo("Info", "No data available for analysis.")
        return

    categories, amounts = zip(*data)
    plt.figure(figsize=(6, 4))
    plt.pie(amounts, labels=categories, autopct='%1.1f%%', startangle=140)
    plt.title("Expense Distribution")
    plt.show()
```

```
# GUI Setup
root = tk.Tk()
root.title("Expense Tracker")

tk.Label(root, text="Amount:").grid(row=0, column=0)
amount_entry = tk.Entry(root)
amount_entry.grid(row=0, column=1)

tk.Label(root, text="Category:").grid(row=1, column=0)
categories = ["Food", "Transport", "Bills", "Shopping", "Entertainment", "Others"]
category_combobox = ttk.Combobox(root, values=categories)
category_combobox.grid(row=1, column=1)

tk.Label(root, text="Description:").grid(row=2, column=0)
description_entry = tk.Entry(root)
description_entry.grid(row=2, column=1)

tk.Label(root, text="Date (YYYY-MM-DD):").grid(row=3, column=0)
date_entry = tk.Entry(root)
date_entry.grid(row=3, column=1)

add_button = tk.Button(root, text="Add Expense", command=add_expense)
add_button.grid(row=4, column=0, columnspan=2)

# Expense Table
columns = ("ID", "Amount", "Category", "Description", "Date")
tree = ttk.Treeview(root, columns=columns, show="headings")
for col in columns:
    tree.heading(col, text=col)
tree.grid(row=5, column=0, columnspan=2)

load_expenses()

analyze_button = tk.Button(root, text="Analyze Expenses", command=analyze_expenses)
analyze_button.grid(row=6, column=0, columnspan=2)

root.mainloop()

# Close DB connection on exit
conn.close()
```