Database Schema

Table for storing transactions:

CREATE TABLE IF NOT EXISTS transactions (
    id INTEGER PRIMARY KEY AUTOINCREMENT,   -- Auto-incremented transaction ID
    type TEXT NOT NULL,   -- Type of transaction: 'Income' or 'Expense'
    date TEXT NOT NULL,   -- Date of transaction in 'YYYY-MM-DD' format
    amount REAL NOT NULL,   -- Amount of the transaction
    category TEXT NOT NULL,   -- Category (e.g., 'Groceries', 'Entertainment')
    description TEXT   -- Optional description of the transaction
);

Table for storing budget limits for categories:

CREATE TABLE IF NOT EXISTS budgets (
    category TEXT PRIMARY KEY,   -- Category name (e.g., 'Groceries', 'Utilities')
    budget_limit REAL NOT NULL   -- Budget limit for the category
);

Sample Data:

To populate the tables with sample data, you can insert the following entries:

Sample transactions:

INSERT INTO transactions (type, date, amount, category, description) VALUES
('Income', '2025-04-01', 3000.00, 'Salary', 'Monthly salary payment'),
('Expense', '2025-04-02', 150.00, 'Groceries', 'Grocery shopping'),
('Expense', '2025-04-03', 100.00, 'Entertainment', 'Movie tickets'),
('Expense', '2025-04-04', 200.00, 'Rent', 'Monthly rent payment');

Sample budgets:

INSERT INTO budgets (category, budget_limit) VALUES
('Groceries', 500.00),
('Entertainment', 300.00),
('Utilities', 150.00),
('Rent', 1000.00);

Explanation of Schema:

1. **transactions**:

   - id: A unique identifier for each transaction (auto-incremented).

   - type: A textual field that specifies the type of transaction: 'Income' or 'Expense'.

   - date: The date of the transaction in 'YYYY-MM-DD' format.

   - amount: The monetary value of the transaction.

   - category: The category the transaction belongs to (e.g., 'Groceries', 'Rent').

   - description: An optional description for more details about the transaction.

2. **budgets**:

   - category: The name of the category (e.g., 'Groceries', 'Utilities').

   - budget_limit: The budget limit for the given category, used for comparison with spending.

Database Setup:

The DatabaseHandler class handles the creation and insertion of data. The code in the provided DatabaseHandler will:

- Initialize the database on the first run by creating the necessary tables.

- Insert and update transactions and budget data.

- Fetch transactions and budget information to manage finances.