PROJECT REPORT

PERSONAL FINANCE TRACKER DATABASE

VARSHITH DONTHA

varshithvarshith814@gmail.com

October 27, 2025

Introduction

The objective of this project was to develop a robust, scalable database schema for a personal finance tracking application. The goal was to create a logical backend structure capable of managing users, categorizing transactions, and recording all income and expenses efficiently. This report outlines the development process, the tools used, and the final structure of the completed database.

Abstract

This project successfully implemented a relational database schema for personal finance management. The final structure consists of four core tables: `Users`, `Categories`, `Income`, and `Expenses`, linked by foreign keys to ensure data integrity. The design supports multi-user functionality, custom categorization, and detailed transaction logging. Key deliverables include the complete SQL schema, a set of dummy data for testing, and several query templates for generating financial reports, such as monthly expense summaries and net balance tracking.

Tools Used

The project was executed using standard, industry-proven technologies:

- **Database System:** MySQL
- **IDE:** MySQL Workbench
- **Language:** SQL (Structured Query Language)
- **Key SQL Features:**
  - Relational table design (Normalization)
  - Foreign Key constraints (for data integrity)
  - `DECIMAL` data type (for financial accuracy)
  - `GROUP BY` clauses (for aggregation)
  - Common Table Expressions (CTEs)

- SQL `VIEW`s (for simplified reporting)

Steps Involved in Building the Project

The project was completed in five distinct phases:

(1) **Schema Design:** The initial phase involved designing the database schema. Four primary entities were identified and normalized: `Users` (to manage individual accounts), `Categories` (a user-defined table for both income and expenses), `Income` (to log incoming funds), and `Expenses` (to log outgoing funds). Relationships were established using `user_id` and `category_id` foreign keys.

(2) **Table Creation:** The logical design was translated into a physical SQL script. `CREATE TABLE` statements were written, paying close attention to correct data types (e.g., `DECIMAL(10, 2)` for currency) and constraints (e.g., `NOT NULL`, `UNIQUE`, `FOREIGN KEY`).

(3) **Data Population:** To validate the schema and test queries, a set of dummy data was created. Two sample users were inserted, along with associated categories and several income/expense transactions for a sample month.

(4) **Query Development:** A suite of SQL queries was developed to serve common reporting needs. This included a monthly expense summary, a detailed category-wise spending report, and a comprehensive net balance summary (Income vs. Expense) using CTEs.

(5) **View Creation:** For ease of access and to simplify reporting, the net balance query was encapsulated into an SQL `VIEW` named `v_monthly_financial_summary`. This allows an application front-end or analyst to retrieve the full monthly balance sheet with a simple `SELECT` query.

Conclusion

The project was completed successfully. The resulting database schema is normalized, efficient, and ready for integration with an application front-end. The developed queries and views provide the necessary tools for essential financial reporting and analysis. The structure is scalable and can support future enhancements, such as budgeting goals or investment tracking.