PERSONAL FINANCE AND HEALTH ANALYZER AND OPTIMIZER

This is one of the finance related MySQL projects which focuses on personal finance analysis and optimization of personal finances. This is one of the niches that is highly relevant on today’s world and is rarely used in depth. The project will be designed using MySQL.

This project is a MySQL based system designed to help individual’s finances. It will include tracking income, expenses, savings, investments and debts. It will provide actionable insights that will help improve financial well-being. The system will use MySQL concepts to generate reports, identify financial trends and suggest optimization strategies.

Prerequisites for this project

In order to perform this project, there are some prerequisites the user needs to have -

1. MySQL 8.0 Command Line Client
2. MySQL 8.0 Workbench CE
3. MySQL Statements
4. MySQL Operations
5. MySQL Clauses
6. MySQL Constraints
7. MySQL Subqueries
8. MySQL Joints
9. User permissions (GRANT & REVOKE)
10. Transactions

These are the concepts or technologies in which the user needs to be fundamentally strong.

Key features of this project

This project will contain several important features of the real-world finance management such as –

1. Income and Expense Tracking: It is used to track monthly income sources and categorize expenses.
2. Savings and Investment Analysis: Monitoring savings account and investment portfolios along with calculating returns.
3. Debt Management: It is used to track debts (credit cards, loans, etc.) and calculate interest payments along with suggesting debt repayment strategies.
4. Financial Health Scoring: It is used to generate a financial health score based on income, expenses, savings and debts. It is used to asses financial stability.
5. Budget Optimization: It is used to suggest budget allocations that are optimal based on previous data.
6. User Permissions: Used to provide role-based access. It is also used to restrict access on sensitive financial data.

Schema

In order to perform this project, there are various parameters required –

1. Database: Create a new database for this project with the project name “Personal\_Finance\_Health”.
2. Tables required:
3. Users – This table should contain user details (user\_id, user\_name, role, password).
4. Income – Used to store income sources of the user (income\_id, user\_id, source, amount, date).
5. Expenses – It is used to store expense details of the user (expense\_id, user\_id, category, amount, date).
6. Savings – Used to store savings details (saving\_id, user\_id, account\_type, amount, date).
7. Investment – Used to store investment details (investment\_id, user\_id, type, amount, return\_rate, date).
8. Debts – Used to store debt details (debt\_id, user\_id, type, amount, interest\_rate, due\_date).
9. Financial\_Health – Used to store financial health scores ( health\_id, user\_id, score, date).
10. Relationships: With the use of constraint keys, the users table must be linked with all the other tables with the use of primary key and foreign key. This is called one to many relationships.
11. Users:
12. Admin – with admin privileges.
13. User – with user privileges.

Implementation

In order to design the structure for this project and generate analysis based on the data, here are the steps of implementation for this project –

**Step 1: Creating database and creating necessary details.**

>show databases;

>CREATE DATABASE Personal\_Finance\_Health;

>use Personal\_Finance\_Health;

>show tables;

>CREATE TABLE USERS(

user\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_name VARCHAR(100) NOT NULL,

role ENUM(‘admin’ , ’user’) NOT NULL,

password VARCHAR(50) NOT NULL, UNIQUE(user\_name));

> CREATE TABLE INCOME(

income\_id INT AUTO\_INCREMENT,

user\_id INT,

source VARCHAR(100),

amount DECIMAL(10,2) NOT NULL,

date DATE, PRIMARY KEY(income\_id),

FOREIGN KEY(user\_id) REFERENCES USERS(user\_id));

> CREATE TABLE EXPENSE(

expense\_id INT AUTO\_INCREMENT,

user\_id INT,

category VARCHAR(100),

amount DECIMAL(10,2) NOT NULL,

date DATE, PRIMARY KEY(expense\_id),

FOREIGN KEY(user\_id) REFERENCES USERS(user\_id));

> CREATE TABLE SAVINGS(

saving\_id INT AUTO\_INCREMENT,

user\_id INT,

account\_type VARCHAR(100),

amount DECIMAL(10,2) NOT NULL,

date DATE, PRIMARY KEY(saving\_id),

FOREIGN KEY(user\_id) REFERENCES USERS(user\_id));

> CREATE TABLE INVESTMENT(

investment\_id INT AUTO\_INCREMENT,

user\_id INT,

investment\_type VARCHAR(100),

amount DECIMAL(10,2) NOT NULL,

return\_rate DECIMAL(5,2) NOT NULL,

date DATE, PRIMARY KEY(investment\_id),

FOREIGN KEY(user\_id) REFERENCES USERS(user\_id));

> CREATE TABLE DEBTS(

debt\_id INT AUTO\_INCREMENT,

user\_id INT,

debt\_type VARCHAR(100),

amount DECIMAL(10,2) NOT NULL,

interest\_rate DECIMAL(5,2) NOT NULL,

due DATE, PRIMARY KEY(debt\_id),

FOREIGN KEY(user\_id) REFERENCES USERS(user\_id));

> CREATE TABLE FINANCIAL\_HEALTH(

health\_id INT AUTO\_INCREMENT,

user\_id INT,

score DECIMAL(10,2) NOT NULL,

date DATE, PRIMARY KEY(health\_id),

FOREIGN KEY(user\_id) REFERENCES USERS(user\_id));