**PERSONAL FINANCE ANALYZER AND**

**OPTIMIZER**

The personal health analyser and optimizer is a SQL project which focuses on personal finance analysis and optimization of the personal finances. This is one of the niche relevant in todays world and is rarely used in debt

**INTRODUCTION:**

This is a MYSQL based project designed to help individuals to analyse and optimize their personal finances it will include tracking income, expenses , savings, investments and debt. It will provide actionable that will help improving financial health. The system will use MYSQL concepts to generate reports, identify financial threats and suggest strategies

**PREREQUISITES FOR THE PROJECT**

In order to perform this project there are some prerequisites that the user needs to have.

1. MYSQL 8.0 command client
2. MYSQL 8.0 workbench
3. MYSQL statements
4. MSQL operations
5. MYSQL clauses
6. MYSQL constraints
7. MYSQL sub-quires
8. MYSQL joins
9. User permissions(GRANT and REVOKE)
10. Transactions

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

**KEY FEATURES**

This project will contain several important features of real world finance management such as.

1. **Income and expense tracking** : it is used to track monthly income sources and categorise the expenses
2. **Savings and investment analysis** : monitoring saving accounts and investment portfolio along with calculating returns.
3. **Debt management** : it is used to track debts(credit cards, loans, etc). and calculate interest payments along with suggesting repayments 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 assess the 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 also used to restrict access on sensitive financial data.

**SCHEMA**

In order to perform this project there are various parameters required.

1. **Data base** : create a new data base for this project with the project name personal\_financial\_health.
2. Tables required:
3. **Users**: this table should contain user details(user\_id, username, role, password)
4. **Income** : this table is used to store income details of the user (income\_id, user\_id, source, amount, date)
5. **Expenses** : it is used to store the expended details of the user(expense,\_id, user\_id, account\_type, amount, date)
6. **Savings**: it is used to store savings details of the user(savings\_id, user\_id, account\_type, amount, date)
7. **Investment** : it is used to store investment details of the user(investment\_id, user\_id, type, amount, return\_date, date)
8. **Debts**: it is used to store debt details of the user(debt\_id, user\_id, type, amount, interest\_rate, due\_date)
9. **Financial\_ health**: it is used to store the financial score of the user(health\_id, user\_id, score, date)
10. **Relationships** : with the use of the constraint keys 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 relationship
11. **Users** : there are user to be assigned one with admin privileges(ALL PREVILEGES) and one with user privileges(SELECT PREVILAGES).

**IMPLEMENTATIONS**

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

**Step 1**: Creating the databased and creating the tables using MySQL statements.

**Step 2**: Inserting values more than 100 rows in all seven tables to perform the analysis. Apart from the auto increment columns all the columns have to be filled with the data

**Table 1 users:** there should be 10 users among those one should be named as admin where as the rest in should be users

**Table 2 income**: for each user there are different Strems of income along with the profit generated they have been assigned to particular user id that matches the user id from the table users

**Table 3 expenses :** this table covers the expenses of all 9 users and it will classify those expenses into various categories.

**Table 4 savings:** This table provides the values for each and every saving medium for each user.

**Table 5 investments :** this table is used to provide the values of each investment done by the user and the return they have received.

**Table 6 debts:** this table is used to provide values that are related to the loan information of each user and how much amount they have loaned and at what interest.

**Step 3:** providing the necessary analytics based on the each requirement.

1. **Calculating monthly income of the each user**: In order to fetch the details for this requirement three tables have to be used(user, income and expenses) all the tables have to be joined.

**Code**: select users.username,sum(income.amount) - sum(expenses.amount) as Net\_monthly\_income from users left join income on users.user\_id = income.user\_id left join expenses on users.user\_id = expenses.user\_id

where income.date between '2023-10-01' and '2023-10-31' group by users.user\_id;

A screenshot of a computer

Description automatically generated

Fig : Net monthly income

1. **Identify high interest debts:**

**Code:** select users.username, debt.type, debt.amount, debt.interest\_rate from debt inner join users on debt.user\_id = users.user\_id where debt.interest\_rate > 10

order by debt.interest\_rate desc;

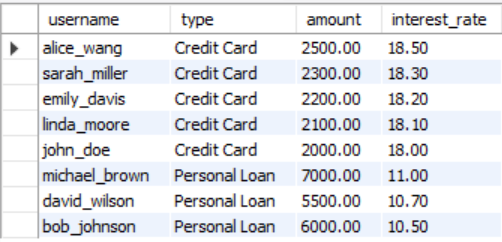


Fig : max interest debts

1. **Generating financial health score**: in this requirement the user will generate the financial health score using users,income, expense, debts

**Code** : select users.user\_id, (sum(income.amount) - sum(expenses.amount) - sum(debt.amount))/ sum(income.amount)\*100 as score, NOW()

from users

left join income on users.user\_id = income.user\_id

left join expenses on users.user\_id = expenses.user\_id

left join debt on users.user\_id = debt.user\_id group by users.user\_id;

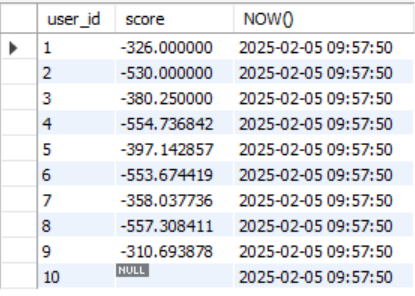


Fig : Financial health score

1. **Budget optimization :**  In this requirement the users will be provided the average spending based on each expenses.

**Code:** select users.user\_id, expenses.category, avg(expenses.amount) as average\_spending from expenses

inner join users on expenses.user\_id = users.user\_id

group by users.user\_id, expenses.category having average\_spending >(select avg(amount) from expenses where category = 'Entertainment');

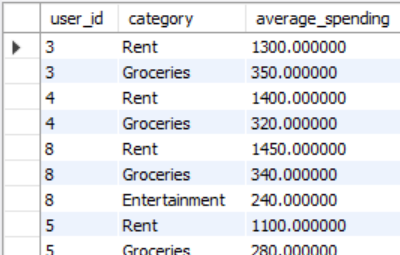


Fig : Average Spending

1. **Calculating savings growth rate:** In this requirement the user is provided with the growth rate using user and savings table.

**Code**: select users.user\_id, savings.account\_type, (savings.amount/(select sum(amount) from savings where user\_id = savings.user\_id))\*100 as savings\_growth\_rate

from savings

inner join users on savings.user\_id = users.user\_id;

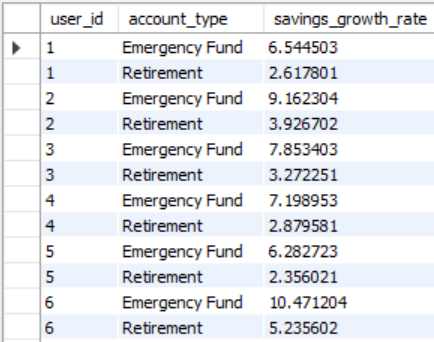


Fig : savings growth rate

**Step 4 - User permissions :** in this step the admin will be provided with all privileges and the user will be provided with only select privileges.

**Step 5 transactions :** Over the course of the period in the data there are several updates when it comes to income, savings, investment, debts and expenses. In order to update the values it is safer to update the values in side the transactions. There are four transactions to be performed on this project.

1. **Record income and update:**

**Code :** start transaction;

select \* from savings;

select \* from income;

savepoint s1;

insert into income (user\_id, source, amount, date) values(1, 'side\_hustle', 9000.00, '2023-10-11');

savepoint s2;

update savings set amount = amount + 9000.00 where user\_id = 1 and account\_type = "retirement fund";

commit;

1. **Debt payoff and update saving:**

**Code:** start transaction;

select \* from savings;

select \* from debt;

savepoint s3;

update savings set amount = amount-1000.00 where user\_id = 1 and account\_type = 'Emergency fund';

savepoint s4;

update debt set amount = amount - 1000.00 where debt\_id = 1;

commit ;

1. **Transfer funds between savings account:**

**Code :** start transaction ;

select \* from savings;

savepoint s5;

update savings set amount = amount-500.00 where user\_id = 1 and account\_type = "emergency\_fund";

savepoint s6;

update savings set amount = amount+500.00 where user\_id = 1 and account\_type = "retirement fund";

commit;

1. **Record investments and update savings:**

**Code**: start transaction;

select \* from savings;

select \* from investment;

savepoint s7;

update savings set amount = amount - 2000.00 where user\_id = 1 and account\_type = "emergency fund";

savepoint s8;

insert into investment(user\_id, type, amount, return\_rate, date) values (1,'mutual\_funds', 2000.00, 6.00, '2023-10-03');

commit;

**CONCLUSION:**

Using this project the users can maintain a good personal finance score hereby helping them while making future financial decisions and maintain a good track of all their future assets and liabilities. With the use of tables such as income, expenses, debts, savings, investments the user generated the financial health table which provided them the scores of the personal finance. Many analytics were produced apart from that many other analytics can be performed based on the data such as finding the debt to income ratio(debt burden analysis, investment performance analysis, monthly expense trends and so fore). The code has to be stored in a .SQL file. Prepare a report of the findings along with the SQL file and it will be posted in git hub.

**FEATURE ENHANCEMENTS :**

The project can be updated with some more features, which are external in nature such as

1. **Integrating with API :** To fetch real life data.
2. **Data visualization :** to provide interactive dashboards and reports
3. **Machine learning :** implementing machine learning models to predict financial health using certain algorithms and provide some recommendations

**IMPACT OF THE PROJECT:**

This project demonstrates proficiency in MYSQL and also the ability to solve real world problems using database systems. It is a unique and practical addition for a portfolio and will help the user advance in the field of database systems.