Database Design Document

Business problem being addressed:

- A centralized system that consists of all the financial data related to investments across various instruments like stocks and mutual funds
- This will enable users to track income, expenses, loans, assets, insurance coverages and liabilities comprehensively
- Goal Setting: Enable users to set financial goals, such as saving for retirement, buying a home, or paying off a debt
- Integrate credit Score tracking and reporting to understand users to improve their creditworthiness
- Calculate the user's current net worth based on the information available related to the assets and liabilities and help them in assessing the overall financial stability
- Generate alerts for important financial events to the users, like credit card bill payment.

Entities

User:

Purpose: The User entity represents the individuals who have created accounts in the application to use the services. This typically stores user-specific information such as their username ,password hash(for authentication) and user id as the unique identifier.

Importance: Users is one of the primary and most important tables in the database, as it is essential for managing user access and personalization.

Relation to Other Entities:

Related to User_Profile: Each logged in user can create multiple profiles, which can be further used to track the profile specific financial information. This establishes a one-to-many relation, where each user can create and manage multiple profiles.

Credit_Info:

Purpose: This entity is used to store the information related to an individual's credit profile. It typically includes data like current credit score, last updated date , unique identifier which identifies credit bureau and unique identifier credit id which uniquely identifies each row.

Importance: Credit information is essential in financial applications for assessing an individual's creditworthiness and managing credit-related aspects

Relation to Other Entities:

Related to User_Profile: Each set of credit information is associated with a specific user profile. The "ProfileID" field in the "Credit_Info" table is a foreign key referencing the "ProfileID" primary key in the "User_profile" table, establishing a one-to-many relationship. This can be used to track credit information for a user from different credit bureau

Related to Credit_Bureau: Credit details can be collected from various credit bureaus, so each record will be associated with the credit bureau from which this data has been collected by using the foreign key credit_bureau_id which is referencing the primary key credit_bureau_id in the credit_bureau table.

Credit_History:

Purpose: The "Credit_History" entity stores historical data related to a user's credit activities. It includes historical records of credit scores and other credit-related events over time.

Importance: Credit history is crucial for assessing an individual's creditworthiness and making informed lending decisions.

Relation to other Entities:

Related to Credit_info: The relationship between these tables is established through the credit_id (FK) field in the "Credit_History" table, which references the credit_id field in the "Credit_Info" table. This relationship allows to associate each credit history entry in the "Credit_History" table with a specific credit record in the "Credit_Info" table. In other words, we can track the history of a user's credit information over time using the "Credit_History" table

Credit_Bureau:

Purpose: The "Credit_Bureau" entity represents credit reporting agencies or bureaus that provide credit information. It includes information about different credit bureaus, such as their names, contact details, address and scale or maximum value of the credit score.

Importance: Credit bureaus are primary sources of credit data, and application uses data from various bureaus to compile credit profiles.

Loans:

Purpose: The "Loans" entity stores information about loans, including details like the loan's unique identifier, the profile or user associated with the loan (via a foreign key), the loan's name, type, the lender's unique identifier and some other details related to the loan.

Importance: Loans are a fundamental aspect of personal finance application, allowing users to manage their borrowing activities, such as mortgages, personal loans, or other types of loans.

Relation to Other Entities:

Related to "Installments": The "Loan_ID" field in the "Installments" table is a foreign key that references the "Loan ID" primary key in the "Loans" table. This establishes a one-to-many relationship, as one loan can have multiple installments.

Related to "Loan_Documents": The "Loan_ID" field in the "Loan_Documents" table is a foreign key that references the "Loan ID" primary key in the "Loans" table, allowing to associate multiple documents with specific loans.

Related to "Lender": The "Lender ID" field in the "Loans" table is a reference to a lender, which can be used to provide information about the loan's origin or source.

Installments:

Purpose: The "Installments" entity records information about individual loan installments, including the installment's unique identifier, the associated loan (via a foreign key), the due amount, due date, payment method the date the installment was paid and amount paid.

Importance: Installments are essential for tracking the repayment of loans and ensuring borrowers meet their payment obligations.

Relation to other entities:

Related to "Loans": The "Loan_ID" field in the "Installments" table is a foreign key that references the "Loan ID" primary key in the "Loans" table, establishing a one-to-many relationship, as one loan can have multiple installments.

Loan_Documents:

Purpose: The "Loan_Documents" entity allows to store documents related to loans. It includes information about the document's unique identifier, the associated loan (via a foreign key), the document's name, URL, and a description.

Importance: Storing loan-related documents is essential for record-keeping and document management in loan applications.

Relation to Other Entities:
Related to "Loans": The "Loan_ID" field in the "Loan_Documents" table is a foreign key that references the "Loan ID" primary key in the "Loans" table, enabling you to link documents to specific loans.

Lender

Purpose: The "Lender" entity represents information about lenders, including their unique identifier, name, type, address, contact information, website, logo, and licensing details.
Importance: Lender information is crucial for providing context and transparency to users regarding the sources of loans and their associated lenders.

Relation to Other Entities:

Related to "Loans": The "Lender ID" field in the "Loans" table is a reference to lenders, allowing to associate loans with specific lenders.

Bank_Information:

Purpose: The "Bank_Information" entity stores information related to bank accounts, including details like the account's unique identifier, the profile of the user associated with the account (via a foreign key), the account type, account number, current balance, and the name of the bank.

Importance: Bank accounts are a core part of personal finance application, allowing users to manage their accounts, track balances, and perform financial transactions.

Relation to Other Entities:

Related to "Bank_Transaction" **:** The "Account_ID" field in the "Bank_Transaction" table is a foreign key that references the "Account_ID" primary key in the "Bank_Information" table. This establishes a one-to-many relationship, as one bank account can have multiple transactions.

Bank_Transaction:

Purpose: The "Bank_Transaction" entity records information about individual bank transactions, including details like the transaction's unique identifier, the associated bank account (via a foreign key), the transaction amount, date, type, and an expense category (via a foreign key).
Importance: Tracking bank transactions is vital for managing personal finances and monitoring income and expenses

Relation to Other Entities:

Related to "Bank_Information"**:** The "Account_ID" field in the "Bank_Transaction" table is a foreign key that references the "Account_ID" primary key in the "Bank_Information" table, establishing a one-to-many relationship, as one bank account can have multiple transactions.

Related to "Expense_Category": The "Expense_cat_id" field in the "Bank_Transaction" table is a foreign key that references the "Expense_cat_id" primary key in the "Expense_Category" table, allowing to categorize transactions.

Expense_Category

Purpose: The "Expense_Category" entity represents expense categories, including details like the category's unique identifier, name, description, and URL.
Importance: Expense categories help users classify and organize their spending, providing insights into their financial habits.

Relation to Other Entities:

Related to "Bank_Transaction": The "Expense_cat_id" field in the "Bank_Transaction" table is a foreign key that references the "Expense_Category_ID" primary key in the "Expense_Category" table. This relationship allows us to categorize transactions based on expense categories.

Related to "Credit_Card_Transaction": The "Expense_cat_id" field in the "Credit_Card_Transaction" table is a foreign key that references the "Expense_Category_ID" primary key in the "Expense_Category" table. This relationship allows us to categorize transactions based on expense categories.

Credit_Card

Purpose: The "Credit_Card" entity stores information about credit cards, including the card's unique identifier, the associated profile or user (via a foreign key), the card issuer, due date, current balance, and credit limit.

Importance: Credit card information is vital for tracking credit card accounts, their balances, and payment due dates.

Relation to Other Entities:
Related to "Credit_Card_Transaction": The "Credit_Card_Id" field in the "Credit_Card_Transaction" table is a foreign key that references the "Credit_Card_Id" primary key in the "Credit_Card" table, establishing a one-to-many relationship, as one credit card can have multiple transactions.

Related to "User_Profile": The "Profile_ID" field in the "Credit_Card" table is a reference to a user's profile, indicating which user owns the credit card.

Credit_Card_Transactions:

Purpose: The "Credit_Card_Transaction" entity records individual credit card transactions, including transaction ID, the associated credit card (via a foreign key), transaction amount, date, transaction type, and an expense category identifier (via a foreign key).

Importance: Tracking credit card transactions helps users monitor their credit card spending and payment obligations.

Relation to Other Entities:

Related to "Credit_Card": The "Credit_Card_Id" field in the "Credit_Card_Transaction" table is a foreign key that references the "Credit_Card_Id" primary key in the "Credit_Card" table, establishing a one-to-many relationship, as one credit card can have multiple transactions.

Related to "Expense_Category": The "Expense_cat_id" field in the "Credit_Card_Transaction" table is a foreign key that links credit card transactions to specific expense categories.

Policy_Details

Purpose: The "Policy_Details" entity stores information about insurance policies, including the policy's unique identifier, the associated profile or user (via a foreign key), the provider (via a foreign key), the type of insurance (via a foreign key), policy number, name, start date, and end date.

Importance: Policy details are fundamental for managing insurance coverage, tracking policy information, and associating policies with specific users.

Relation to other entities:

Related to "Additional_Policy_Details": The "Policy_ID" field in the "Additional_Policy_Details" table is a foreign key that references the "Policy_ID" primary key in the "Policy_Details" table, allowing us to store additional details specific to each policy.

Related to "Insurance_Type": The "Type_id" field in the "Policy_Details" table is a foreign key that references the "Type_ID" primary key in the "Insurance_Type" table, indicating the type of insurance for each policy.

Related to "Claim_History": The "Policy_ID" field in the "Claim_History" table is a foreign key that associates claims with specific insurance policies.

Related to "Insurance_Providers": The "Proider_id" field in the "Policy_Details" table is a foreign key that references the "Type_ID" primary key in the "Insurance_Providers" table, indicating the insurance provider for each policy.

Additional_Policy_Details:

Purpose: The "Additional_Policy_Details" entity allows to store specific details related to each insurance policy, such as coverage limit, premium amount, deductible amount, payment frequency, and next payment date.

Importance: These additional details provide a comprehensive view of the insurance policies, including financial and payment-related information.

Relation to Other Entities:

Related to "Policy_Details": The "Policy_ID" field in the "Additional_Policy_Details" table is a foreign key that references the "Policy_ID" primary key in the "Policy_Details" table, allowing to link additional policy details to specific insurance policies.

Insurance_Type:

Purpose: The "Insurance_Type" entity defines the types of insurance available in the application, including a unique identifier, name, and description for each insurance type.
Importance: Categorizing and describing different insurance types helps users understand their coverage and select the appropriate type of insurance.

Relation to Other Entities:

Related to "Policy_Details": The "Type_id" field in the "Policy_Details" table is a foreign key that references the "Type_ID" primary key in the "Insurance_Type" table, indicating the type of insurance associated with each policy.

Claim_History:

Purpose: The "Claim_History" entity records information about insurance claims, including a unique identifier, the associated insurance policy (via a foreign key), claim amount, and claim date.

Importance: Claim history is essential for tracking and managing insurance claims, providing a record of when and how claims were made.

Relation to Other Entities:

Related to "Policy_Details": The "Policy_ID" field in the "Claim_History" table is a foreign key that associates claims with specific insurance policies.

Insurance_Providers:

Purpose: The "Insurance_Providers" entity contains information about insurance providers or companies, including a unique identifier, name, and description.

Importance: Knowing the providers of insurance policies helps users understand their insurance sources and providers' characteristics.

Relation to Other Entities:

Related to "Policy_Details": The "Proider_id" field in the "Policy_Details" table is a foreign key that references the "Type_ID" primary key in the "Insurance_Providers" table, indicating the provider for each insurance policy.

Financial_Goals

Purpose: The "Financial_Goals" entity stores information about financial goals set by users. It includes details like the goal's unique identifier (Goal_ID), the associated profile or user (Profile_ID), the goal name, description, target amount, target date, and start date.

Importance: Financial goals are a important feature of personal finance applications, enabling users to set, track, and achieve specific financial objectives.

Relation to Other Entities:

Related to "Goal_Progress": The "Goal_ID" field in the "Goal_Progress" table is a foreign key that references the "Goal_ID" primary key in the "Financial_Goals" table. This establishes a one-to-many relationship, as each financial goal can have multiple progress entries tracking its achievement.

Goal_Progress:

Purpose: The "Goal_Progress" entity is used to track the progress made toward financial goals. It includes details like the progress entry's unique identifier (Progress_ID), the associated goal (via a foreign key), the date of the progress entry, the amount achieved, and the current amount toward the goal.

Importance: Tracking progress helps users monitor their journey toward financial goals, stay motivated, and make necessary adjustments to their financial plans.

Relation to Other Entities:

Related to "Financial_Goals": The "Goal_ID" field in the "Goal_Progress" table is a foreign key that links each progress entry to a specific financial goal.

Notification:

Purpose: The "Notification" entity is designed to store information about notifications generated by application. This information may include a unique notification identifier (notification_id), the associated profile or user (profile_id), the date when the notification was created, the type of notification (e.g., payment reminder, transaction alert), the due date related to the notification, and potentially an associated financial amount.

Importance: Notifications are a vital component of financial applications, providing users with reminders, alerts, and important information regarding their financial activities. This can include reminders for bill payments, upcoming financial events, account balance alerts, and more.

Relation to Other Entities:

Related to "Profile": The "profile_id" field in the "Notification" table is a foreign key that references the user's profile, indicating which user the notification is associated with. This allows notifications to be linked to specific users.

Trading_Account:

Purpose: The "Trading_Account" entity stores information related to a user's trading account, including a unique trading account identifier (Trading_Account_Id), the associated user profile (via a foreign key), the user's username for the trading account, the last updated date, and the broker's unique identifier.

Importance: Trading accounts are central to financial applications that involve investing and trading in assets like stocks and securities. This entity allows users to manage and track their trading accounts.

Relation to Other Entities:

Related to "Assets": The "Trading_Account_Id" field in the "Assets" table is a foreign key that associates assets with specific trading accounts.

Assets:

Purpose: The "Assets" entity stores information about assets held in a trading account, such as stocks or securities. It includes details like a unique asset identifier (Asset_ID), the associated trading account (via a foreign key), asset name, quantity held, average value, payment type(SIP/One-Time), and investment type.

Importance: Assets are the core holdings within a trading account. This entity allows users to track users' asset portfolios.

Relation to Other Entities:
Related to "Trading_Account": The "Trading_Account_Id" field in the "Assets" table is a foreign key that links assets to specific trading accounts.

Related to "Purchase_Details": The "Asset_ID" field in the "Purchase_Details" table is a foreign key that associates purchase details with specific assets, this is used to track transactions associated with an assett.

Related to "Sip_Details": The "Asset_ID" field in the "Sip_Details" table is a foreign key that links systematic investment plans (SIP) to specific assets.

Purchase_Details

Purpose: The "Purchase_Details" entity is used to record purchase details for assets, including a unique purchase detail identifier (ID), the associated asset (via a foreign key), purchase date, buy/sell type, quantity, and price.

Importance: Purchase details are essential for tracking individual transactions, such as buying and selling assets.

Relation to Other Entities:

Related to "Assets": The "Asset_ID" field in the "Purchase_Details" table is a foreign key that associates purchase details with specific assets.

Sip_Details:

Purpose: The "Sip_Details" entity is designed to store information about systematic investment plans (SIPs), including a unique SIP identifier (Sip_ID), start date, the associated asset (via a foreign key), payment frequency, payment amount, and the next payment date.

Importance: SIPs are a common investment strategy in which regular payments are made into assets over time. This entity allows users to set up and manage SIPs.

Relation to Other Entities:

Related to "Assets": The "Asset_ID" field in the "Sip_Details" table is a foreign key that associates SIPs with specific assets.

User_profile:

Purpose: The "User_profile" entity serves as the primary entity for storing user profiles, including critical user details such as a unique profile identifier (Profile_Id), user ID (User_id), first name (First_Name), last name (Last_Name), email address (Email), date of birth (Dob), parent profile identifier (Parent_Profile_ID), and Social Security Number (SSN_No).

Importance: User profiles are at the core of the application, and this entity holds user-specific information, allowing users to interact with application and access personalized features.