

MintTrack - Stage 2: Database Design

Conceptual Design (ERD)

For the MintTrack project, we created an **Entity-Relationship Diagram (ERD)** to represent the conceptual design of the database. The ERD diagram includes the essential entities and relationships for tracking shared expenses, bill splitting, and notifications.

Relationships

1. User—Group (Many-to-Many)

- **Explanation:** A user can belong to multiple groups, and each group can have multiple users. The UserGroup join table links users and groups.
- **Implementation:** UserGroup is the table that connects users to the groups they belong to.
- **Cardinality:** Many-to-Many.

2. Group—Transaction (One-to-Many)

- **Explanation:** Each group can have multiple transactions, but a transaction is associated with one group.
- **Cardinality:** One-to-Many.

3. Transaction—User (Many-to-Many)

- **Explanation:** Each transaction involves at least two users (a sender and a receiver), and each user can participate in multiple transactions.
- **Cardinality:** Many-to-Many.

4. Transaction—Spending (One-to-Many)

- **Explanation:** A transaction can involve multiple spending entries, as a transaction might include multiple categories of spending (e.g., food, travel).
- **Cardinality:** One-to-Many.

5. Transaction—CurrencyExchange (Many-to-One)

- **Explanation:** Each transaction may involve a currency exchange, but a currency exchange rate is valid for many transactions during a specific time window.
- **Cardinality:** Many-to-One.

Assumptions for Each Entity

- **User:**

Assumptions: Each user must register in the system to participate in groups and manage expenses. Users can belong to multiple groups and will receive notifications about transactions via their registered phone number. The DateCreated attribute captures when the user registered in the system.

Reasoning: The User entity is required to manage individual users' details and track their participation in the system. The phone number is important for sending payment reminders and notifications, while DateCreated helps track account activity and history.

- **Group:**

Assumptions: Groups are created by users to organize shared expenses. Users can create or join multiple groups, and each group is identified by a unique GroupId. The CreatedBy attribute ensures that each group has a responsible creator, and the CreateAt and DeleteAt timestamps track the lifecycle of each group.

Reasoning: The Group entity allows users to manage expenses specific to a set of participants. Tracking the creator of a group and its creation/deletion times helps manage group dynamics and audit group activities.

- **Transaction:**

Assumptions: Transactions represent shared expenses within a group. Each transaction is associated with two users: the sender and the receiver, who are linked via SenderId and ReceiverId. The GroupId connects the transaction to a specific group, while Amount, CurrencyType, and Date capture the financial details of the expense.

Reasoning: The Transaction entity ensures accurate logging of expenses between users within a group. The inclusion of SenderId and ReceiverId helps track who initiated the transaction and who is responsible for the payment. Recording the currency type is essential for handling multi-currency transactions.

- **Spending:**

Assumptions: Each transaction is categorized into a specific type of spending (e.g., food, rent), captured by the Category attribute. The amount spent is

recorded in a specific currency (CurrencyType). This entity breaks down each transaction into its spending components for more detailed financial tracking.

Reasoning: The Spending entity supports finer categorization and analysis of expenses within a transaction. This entity is important for generating reports and analytics based on different categories of spending, allowing users to understand their financial behavior better.

- **CurrencyExchange:**

Assumptions: Currency exchange rates are necessary for converting transaction amounts when expenses are logged in different currencies. The SourceCurrency and TargetCurrency attributes define the currency pair, and the Rate provides the exchange value at a particular Timestamp.

Reasoning: The CurrencyExchange entity is crucial for handling multi-currency transactions, ensuring accurate conversion between currencies at the time of the transaction. This enables the application to support users operating across different currencies.

- **Inflation**

Assumptions: Inflation data, captured through the Year, Month, and Rate attributes, is used to contextualize financial data over time. This information can be applied to financial reports or projections to adjust for the effect of inflation on expenses.

Reasoning: The Inflation entity helps to incorporate economic data into the application, allowing for more accurate long-term financial planning. Users can view their expenses in real terms, adjusted for inflation over time.

Normalization

The database schema adheres to **Third Normal Form (3NF)** by ensuring:

- **First Normal Form (1NF):** All attributes have atomic values.
- **Second Normal Form (2NF):** All non-primary key attributes are fully dependent on the primary key.
- **Third Normal Form (3NF):** There are no transitive dependencies, ensuring that all non-key attributes are only dependent on the primary key.

Relational Schema

Here is the logical design of the relational schema based on the ERD:

1. User

User(UserId: INT [PK], Name: VARCHAR(100), PhoneNumber: VARCHAR(15),
DateCreated: DATE)

2. Group

Group(GroupId: INT [PK], GroupName: VARCHAR(100), CreatedBy: INT [FK to
User.UserId], CreateAt: DATE, DeleteAt: DATE)

3. Transaction

Transaction(TransactionId: INT [PK], GroupId: INT [FK to Group.GroupId], Amount:
DECIMAL(10,2), CurrencyType: VARCHAR(3), Date: DATE, SenderId: INT [FK to
User.UserId], ReceiverId: INT [FK to User.UserId])

4. Spending

Spending(SpendingId: INT [PK], CurrencyType: VARCHAR(3), Category: VARCHAR(50),
Amount: DECIMAL(10,2))

5. CurrencyExchange

CurrencyExchange(Timestamp: DATETIME [PK], SourceCurrency: VARCHAR(3),
TargetCurrency: VARCHAR(3), Rate: DECIMAL(10,6))

6. Inflation

Inflation(Year: Integer[PK], Month: Integer[PK], rate: Real)

Conclusion

This stage of the project includes a comprehensive database design with normalized tables and clearly defined relationships. The ERD captures all necessary components of the MintTrack system for managing shared expenses, bill splitting, and notifications. The schema adheres to normalization standards to ensure data integrity and efficiency.