

Table Names, Columns, Keys (The starred* portions)

- User (UserID, Email, FirstName, LastName, EncryptedPassword, UnallocatedFunds, CreatedAt, UpdatedAt)
- Income (IncomeID, UserID, Amount, IncomeDate, Description, CreatedAt, UpdatedAt)
 - Foreign Key UserID references User
- Category(CategoryID, UserID, Name, MonthlyDefault, Amount, CreatedAt, UpdatedAt)
 - Foreign Key UserID references User
- Expense (ExpenseID, CategoryID, Amount, ExpenseDate, Description, CreatedAt, UpdatedAt)
 - Foreign Key CategoryID references Category

Explanation of Table Names, Column Names

- User - a table with records created for every user who creates an account in the app.
 - UserID: A unique identifier for the user (auto-generated)
 - Email: The user's email address
 - FirstName: The first name of the user
 - LastName: The last name of the user
 - EncryptedPassword: The password that the user creates (encrypted for safety)
 - UnallocatedFunds: The amount of money that the user has that they need to allocate to categories
 - CreatedAt: A timestamp of when the record was created
 - UpdatedAt: A timestamp of when the record was last updated
- Income - a table with records identifying income that any given user receives (money added to their account).
 - IncomeID: A unique identifier for the income (auto-generated)
 - UserID: A foreign key to reference which user the income is for
 - Amount: The dollar amount of the income
 - IncomeDate: The date that they received the income
 - Description: A text description the user provides for where that income came from
 - CreatedAt: A timestamp of when the record was created
 - UpdatedAt: A timestamp of when the record was last updated
- Category - a table with records identifying the different expense categories a user may create (e.g., rent, groceries, etc.)
 - CategoryID: A unique identifier for the category(auto-generated)
 - UserID: A foreign key to reference which user the category belongs to
 - Name: The textual name of the expense category
 - MonthlyDefault: A dollar amount that represents the default amount of money needed for the given expense categories in a normal month

- Amount: the total amount of money the user has allocated to this expense category that hasn't been removed from expenses
 - CreatedAt: A timestamp of when the record was created
 - UpdatedAt: A timestamp of when the record was last updated
- Expense - a table with records identifying a single expense of a given user.
 - ExpenseID: A unique identifier for the expense (auto-generated)
 - CategoryID: A foreign key to reference which expense category the expense was for
 - Amount: The dollar amount of the expense
 - ExpenseDate: The date of the expense transaction
 - Description: A description of where that money was spent
 - CreatedAt: A timestamp of when the record was created
 - UpdatedAt: A timestamp of when the record was last updated

Entity Representations

- A user represents a person who is using the application and desires to manage their budget using a zero-balance budget system.
- An income represents a paycheck or other increase in a person's wealth. For example, this may be something they sold on Craigslist, or simply their bi-weekly paycheck.
- A category represents a portion of a person's planned expenses. If a budget refers to the allocation of a person's entire wealth, a category is one section of the person's budget.
- An expense represents a purchase or other decrease in a person's wealth.

Relations

- The Expense entity/table is assigned to a category, and as such holds a foreign key to the Category entity/table.
 - There are many Expenses to one Category.
- The Category entity/table belongs to a specific user, and as such holds a foreign key to the User entity/table.
 - There are many Categories to one User.
- The Income entity/table is earned by a user, and as such holds a foreign key to the User entity/table.
 - There are many Incomes to one User.

Evidence of Normalization

- All of our tables meet the criteria to be in 4th Normal Form:
 - 1NF - No single attribute contains a list of values.
 - While a description (as seen in the income/expense relations) provided by a user could possibly contain multiple values within it, it is still considered

just a single comment used to describe extra details for an income/expense.

- 2NF - 2nd Normal Form can only be violated when there are composite keys. We have no composite keys and thus pass 2NF.
- 3NF - Every determinant of the functional dependencies in our tables is a candidate key for the relation
- 4NF - Our tables are in 3NF and they have no multivalued dependencies