Expense Tracker System

Logical Database Design Report

October 28, 2025

Contents

1. Introduction

This document presents the logical database design of the **Expense Tracker System**. The design follows a relational model and implements the required functionalities:

- Add, edit, and delete expenses.
- Add notes and descriptions to expenses.
- Categorize expenses.
- Set and manage monthly budgets per category.
- Customize default categories.
- View remaining budget per category.

The design is normalized, relational, and suitable for offline desktop use.

2. Database Schema (DBML Representation)

```
Table Currency {
  currency_code char(3) [pk]
  symbol varchar(5)
  name varchar(50)
}

Table User {
  user_id int [pk, increment]
  full_name varchar(100)
  email varchar(100) [unique]
  password_hash varchar(255)
  currency_code char(3)
  created_at datetime
}

Table Category {
  category_id int [pk, increment]
  user_id int
  name varchar(50)
  description varchar(255)
  is_default boolean
  color_code varchar(7)
}

Table Expense {
  expense_id int [pk, increment]
  user_id int
  category_id int
  amount decimal(10,2)
  expense_date date
  description text
  note varchar(255)
  created_at datetime
  updated_at datetime
}

Table Budget {
  budget_id int [pk, increment]
  user_id int
  category_id int
  month int
  year int
  amount_limit decimal(10,2)
  created_at datetime
  updated_at datetime
  Indexes {
    (user_id, category_id, month, year) [unique]
  }
}

Ref: User.currency_code > Currency.currency_code
Ref: Category.user_id > User.user_id
Ref: Expense.user_id > User.user_id
Ref: Expense.category_id > Category.category_id
Ref: Budget.user_id > User.user_id
Ref: Budget.category_id > Category.category_id
```

3. Detailed Table Explanations

3.1. 1. User Table

Purpose: Stores user account data and preferences. **Attributes:**

- **user_id (PK):** Unique identifier for each user. Auto-incremented.
- **full_name:** The full name of the user.
- **email:** User's unique email address (acts as a candidate key).
- **password_hash:** Securely stored password hash.
- **currency_code (FK):** Links to the *Currency* table.
- **created_at:** Timestamp when the account was created.

Super keys: {user_id}, {email}. **Candidate keys:** {user_id}, {email}. **Foreign keys:** currency_code → Currency(currency_code).

3.2. 2. Category Table

Purpose: Defines expense categories (default or custom). **Attributes:**

- **category_id (PK):** Unique ID for each category.
- **user_id (FK):** References the owner in the *User* table.
- **name:** Category name (e.g., Food, Transport, Rent).
- **description:** Optional description for the category.
- **is_default:** Indicates if category is system-defined or user-defined.
- **color_code:** Visual identifier (e.g., HEX color).

Super keys: {category_id}, {user_id, name}. **Foreign keys:** user_id → User(user_id).

3.3. 3. Expense Table

Purpose: Stores individual expenses linked to a user and category. **Attributes:**

- **expense_id (PK):** Unique identifier for each expense.
- **user_id (FK):** Owner of the expense.
- **category_id (FK):** Category assigned to the expense.
- **amount:** Amount spent.

- **expense_date:** Date the expense occurred.
- **description:** Expense description.
- **note:** Additional notes.
- **created_at / updated_at:** Timestamps for creation and modification.

Super keys: {expense_id}. **Foreign keys:**

- user_id → User(user_id)
- category_id → Category(category_id)

3.4. 4. Budget Table

Purpose: Tracks monthly budgets for each category. **Attributes:**

- **budget_id (PK):** Unique identifier for each budget entry.
- **user_id (FK):** User who owns the budget.
- **category_id (FK):** Category assigned to this budget.
- **month / year:** Time period of the budget.
- **amount_limit:** Spending limit.
- **created_at / updated_at:** Management timestamps.

Super keys: {budget_id}, {user_id, category_id, month, year}. **Foreign keys:**

- user_id → User(user_id)
- category_id → Category(category_id)

3.5. 5. Currency Table

Purpose: Stores supported currencies for users. **Attributes:**

- **currency_code (PK):** ISO currency code (e.g., USD, EUR).
- **symbol:** Currency symbol ($, €, £, etc.).
- **name:** Full currency name.

Super keys: {currency_code}.

4. Relationships Summary

- **User–Currency (1:N):** Each user selects one currency; one currency may be used by many users.
- **User–Category (1:N):** Each user can define multiple categories.
- **User–Expense (1:N):** Each expense belongs to one user.
- **User–Budget (1:N):** Each user can define multiple budgets.
- **Category–Expense (1:N):** Each category may contain multiple expenses.
- **Category–Budget (1:N):** Each category can have a monthly budget entry.

5. Normalization Summary

All tables are designed to satisfy:

- **1NF:** All fields hold atomic values.
- **2NF:** Non-key attributes depend fully on the primary key.
- **3NF:** No transitive dependencies exist.

6. Conclusion

The Expense Tracker System database is a fully normalized relational design supporting expense management, categorization, and budgeting functionalities. This model can be implemented in relational DBMSs such as MySQL, SQLite, or PostgreSQL, suitable for offline desktop systems.