Smart Spending Tracker
Personal Expense & Budget Management System

Author: Aman E

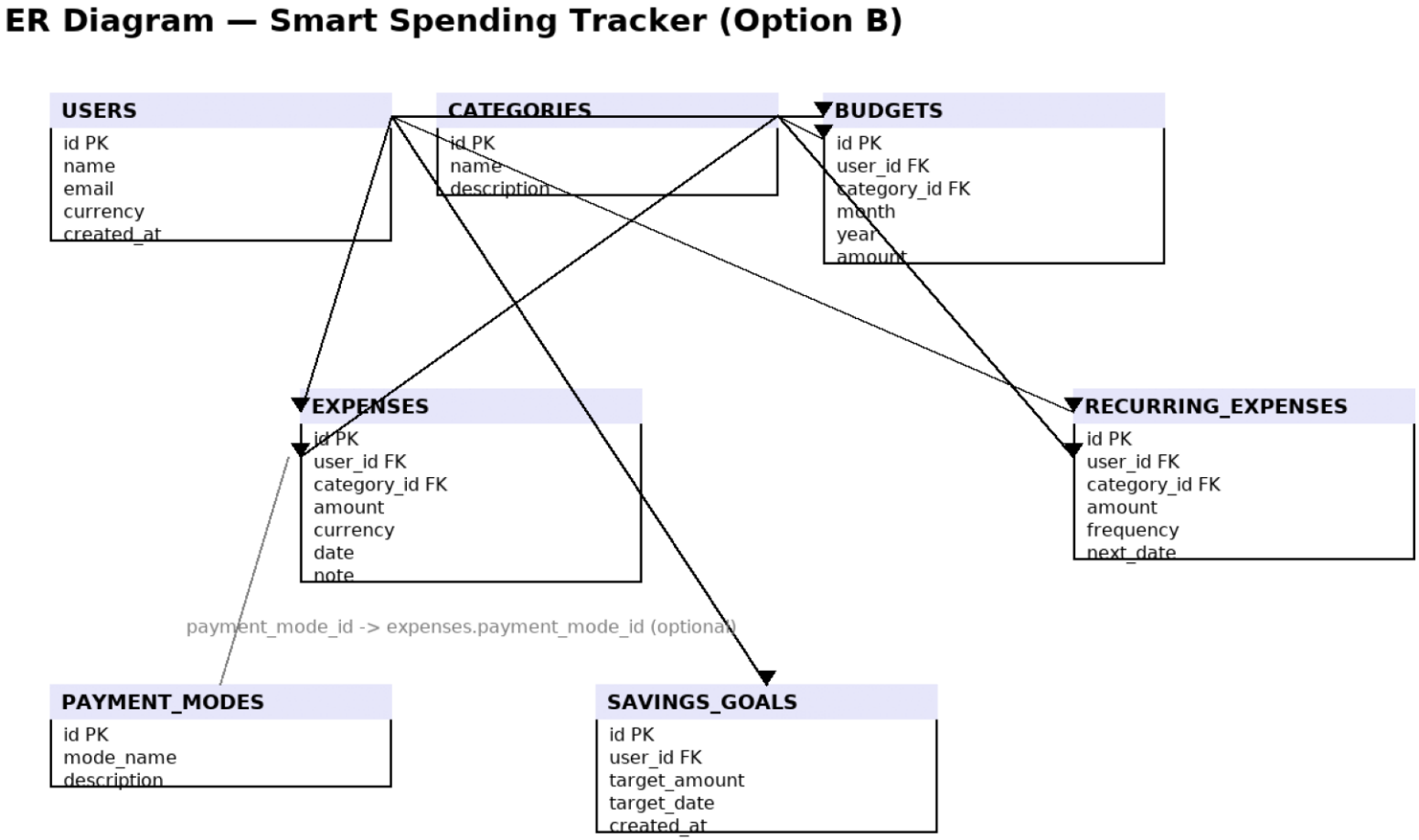
Date: November 21, 2025

Tech stack: PostgreSQL (schema), Python (Flask + SQLAlchemy), Power BI

Database Schema — Option B (Advanced)

Tables included: - users: id PK, name, email, currency, created_at - categories: id PK, name, description - budgets: id PK, user_id FK, category_id FK, month, year, amount - expenses: id PK, user_id FK, category_id FK, amount, currency, date, note - payment_modes: id PK, mode_name, description - savings_goals: id PK, user_id FK, target_amount, target_date, created_at - recurring_expenses: id PK, user_id FK, category_id FK, amount, frequency, next_date Highlights: - Trigger function to auto-create alerts when expenses exceed budgets. - Unique constraint on budgets(user_id, category_id, month, year) to avoid duplicates. - Use of numeric types for monetary fields for precision.

ER Diagram (Option B)



Key Features & API Endpoints

- SQL: schema.sql, insert_sample_data.sql, triggers_and_functions.sql (budget exceed alerts)
- Python: Flask API (endpoints: POST /api/expenses, GET /api/users/<id>/summary)
- Power BI: dashboard connecting to exported CSV or direct DB query for visuals
- Sample data included for quick demos and screenshots

How to run locally

1. Install PostgreSQL, create database 'spendtracker' and run SQL scripts in database/

2. Create a Python virtual environment and install: pip install -r python-app/requirements.txt

3. Set DATABASE_URL env var and run: python python-app/main.py

4. Use Postman or sample_requests.http to add expenses and test alerts

5. Open Power BI Desktop and import analytics/sample_data_for_powerbi.csv to build dashboard

Author: Aman E • Generated on November 21, 2025