Habitat for Humanity LA: Database System Summary

Volunteer, Project, and Donation Management

Yea Yen Kim

Fall 2024

NOTE: This is a summary document. For more details, please see docs/HabitatLA_Tech_Design

1. Project Overview

This database project supports Habitat for Humanity Los Angeles (Habitat LA) – a nonprofit organization that builds and repairs affordable housing for low-income families. Habitat LA coordinates volunteers and donors across housing and community projects in the LA area.

This database system focuses on the following operational data:

- Volunteer registration and scheduling
- Project planning and assignments
- Donation tracking and allocation

2. Core Features

Volunteer Management

- Tracks contact info, availability, and roles
- Assigns volunteers to projects based on availability
- Logs hours served on each project

Project Management

- Stores project name, location, description, start/end dates
- Tracks volunteers assigned to each project
- Connects projects with donations received

Donation Management

- Records donor info and donation amounts
- Allocates donations across multiple projects
- Tracks the purpose of each donation (e.g., "Building Materials")

3. Database Structure

Main Tables & Relationships:

- Volunteer: Personal info (Name, Email, Phone, Location)
- VolunteerAvailability: Links each volunteer to available weekdays and time slots
- Role: Predefined volunteer roles (e.g., Builder, Planner)
- VolunteerAssignment: Connects volunteers to projects, with assigned roles and time availability
- ServiceReports: Logs hours served per volunteer per assignment
- Project: Metadata about projects (Name, Location, Dates)
- Donation: Date and amount of each donation, linked to donor
- Donor: Donor name and contact info
- DonationProject: Maps donation amounts to specific projects and purposes

These entities are normalized and joined via foreign keys.

4. Technologies / Features Used

- **MySQL** – Relational database to store, organize, and query entity relationships.
- **Python** – Used for data migration, validation, insertion, error logging, dry run testing, and automation scripts.
- **Modular File Structure** – Organized migration and insert logic across multiple Python folders and files for maintainability.
- **Custom Logging System** – Logs all migration operations and errors.
- **Dry Run Support** – Enables testing of migrations without affecting the database, useful for debugging.
- **Queries** – Advanced queries such as Joins, Grouping, Aggregation, Filtering for insights.

5. Skills Learned

Through this project, I gained practical, hands-on experience with the full lifecycle of relational database development. Here are some key takeaways:

- **Relational Database Modeling**

  Developed entity-relationship diagrams (ERDs), wrote business rules, normalized schemas to 3NF, and translated those designs into SQL to build a functioning MySQL database.

- **Business Logic & Query Design**

  Wrote and tested a variety of meaningful queries using SQL joins, aggregations, filtering, and grouping.

- **Data Migration & Validation**

  Built a custom Python-based migration pipeline. Integrated validation (e.g., email/phone/date formats, duplicate checking) and logging for clean and traceable data insertions.