Gym Management Database Design Document

Course: Harvard CS50's Introduction to Databases with SQL

Database Engine: SQLite

Project Author: Samuel Levin

1. Introduction

This document describes the design of a relational database system that models the operations of a fitness gym. The database supports membership management, staff assignments, class scheduling, billing, and attendance tracking. It is implemented in SQLite and accessed via automated batch files that execute predefined SQL scripts.

1.1 Objective

The purpose of this document is to define the database schema, constraints, and data access design for the Gym Management System. It provides a foundation for implementation, review, and maintenance.

1.2 Intended Audience

This document is intended for technical reviewers, database developers, and stakeholders who require insight into the logical and physical design of the gym database system.

2. System Overview

The Gym Management Database serves as the core data store for all operational and analytical needs of the gym. It supports both daily transactions (e.g., class enrollments, check-ins) and analytical reporting (e.g., monthly revenue, member retention, class utilization). The database interacts with user-friendly batch files that automate data queries and updates.

3. Design Considerations

Key design considerations include:

- Maintain historical membership data for retention analysis.
- Enforce referential integrity through foreign key constraints.
- Prevent duplicate enrollments via unique constraints.
- Ensure scalability through proper indexing on common query paths.

4. Logical Design

The database schema models gym operations using normalized tables that capture members, memberships, staff, classes, schedules, enrollments, transactions, and attendance. The design adheres to Third Normal Form (3NF) to eliminate redundancy and ensure data integrity.

Key Entities:

- members – Member demographic and contact details.
- memberships – Catalog of available membership plans.
- member_membership_history – Tracks membership history with start and end dates.
- staff – Gym employees including instructors and administrators.
- rooms – Physical locations where classes take place.
- classes – Template definitions for fitness classes.
- schedules – Links classes, rooms, and instructors to specific time slots.
- class_enrollments – Tracks which members are enrolled in each class schedule.
- attendance – Member check-ins and visit durations.
- transactions – Financial transactions for memberships and other services.

5. Physical Design

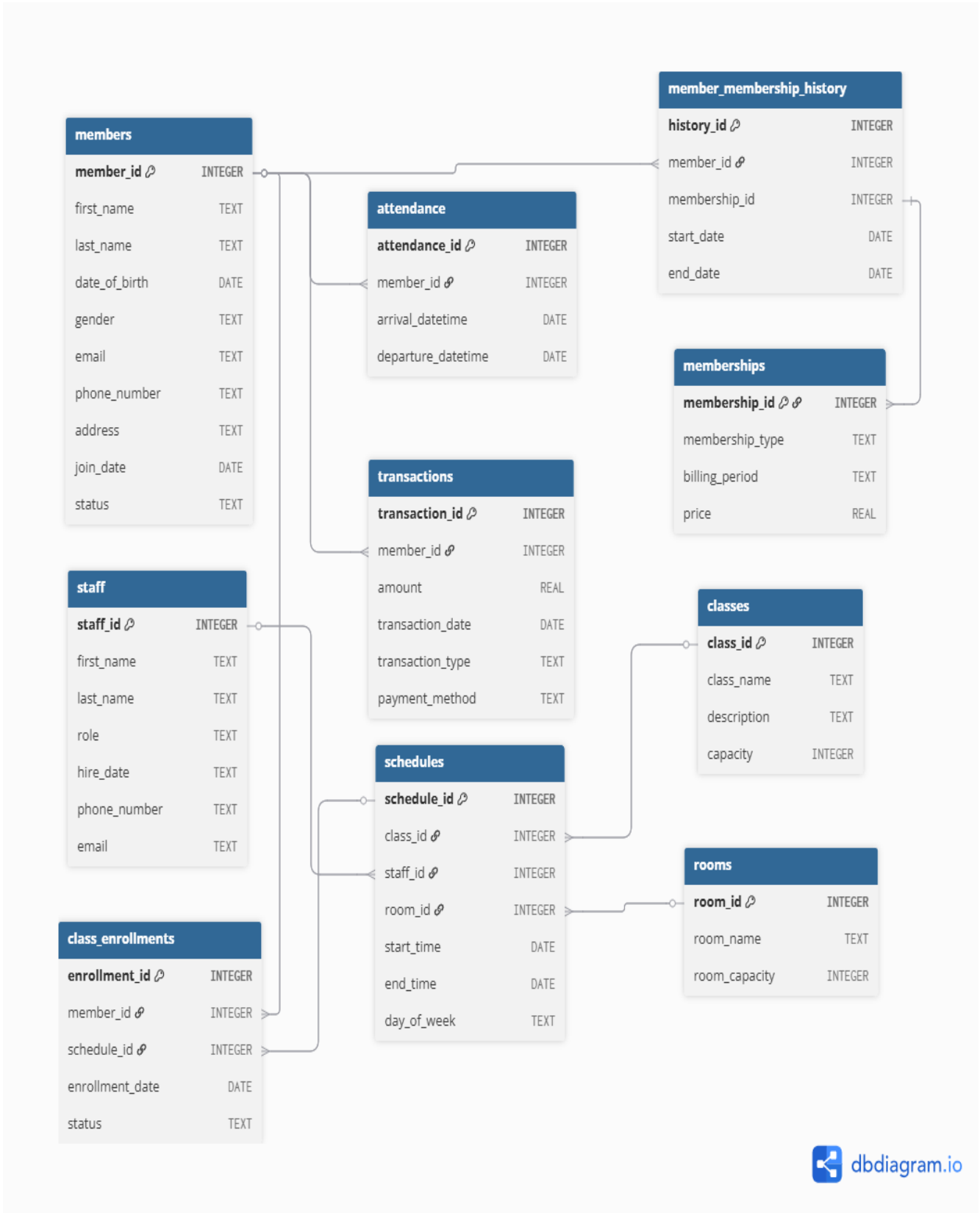
The database enforces integrity through the following constraint mechanisms:

- FOREIGN KEY constraints with ON DELETE actions.
- CHECK constraints to control status and transaction type values.
- UNIQUE constraints on member and staff emails.
- Composite UNIQUE(member_id, schedule_id) in class_enrollments to avoid double booking.

Indexing Strategy:

- Composite indexes on member_id/date for transactions.
- Indexes on enrollment status and schedule_id for faster reporting.

6. Data Dictionary Diagram



7. Accessing SQLite Environment

1. **Install an IDE (Integrated Development Environment)**
    - Download and install Visual Studio Code or another IDE of your choice (learn more about IDEs here).
2. **Download the Project from GitHub**
    - Go to my project's GitHub repository. You should be in Slev1200/sql-database-projects
    - Click the green **"Code"** button and select **"Download ZIP"**.
    - Once downloaded, **unzip** the folder to extract the project files.
3. **Open the Project in Your IDE**
    - Launch your IDE and navigate to **File → Open Folder**.
    - Select the unzipped project folder and open it.
4. **Navigate to the Project Directory**
    - Open a **terminal** inside your IDE (in VS Code, use *View → Terminal*).
    - Change the directory to where your gym.db file is located. For example:
    - cd path/to/your/project/folder
5. **Access the SQLite Environment**
    - Once inside the correct directory, type the following command to start SQLite: **sqlite3 gym.db**
    - You will now be inside the SQLite environment where you can write and execute SQL queries directly against your database.

8. Batch File Interface Instructions

Several Windows batch script tools were implemented to simplify user interaction with the database. Each script calls SQLite with prewritten SQL queries and formatted output.

To see help for a specific tool, run **tool.bat /?**. **(**E.g.
Member.bat /?)

- **all_mem.bat –** Find a list of members in members table

  Usage: pass one argument: num_members
  Description: This batch file provides a list of members of the club.
    - Example: all_mem.bat 3

first_name	last_name	date_of_birth	gender	email	phone_number	address	join_date	status
Emma	Rodriguez	1990-05-14	F	emma.rod@example.com	6175551001	12 Oak St, Boston, MA	2025-11-01	active
Liam	Johnson	1988-09-22	M	liam.j@example.com	6175551002	77 Main St, Cambridge, MA	2025-11-01	active
Olivia	Nguyen	1995-03-02	F	olivia.n@example.com	6175551003	45 Beacon St, Boston, MA	2025-11-01	active

- **member.bat –** Find a specific member in members table
  Usage: member.bat first_name last_name
  Description: This batch file retrieves member data for the specified member. You must provide exactly two string arguments.

- Example: member.bat Samuel Lenin

member_id	first_name	last_name	date_of_birth	gender	email	phone_number	address	join_date	status	current_payment_type
52	Samuel	Lenin	2000-05-14	M	samuel.len@example.com	9781234567	13 Oak St, Boston, MA	2025-11-02	active	Card

- **all_cl.bat –** List out however many classes you want to look at

  Usage: all_cl.bat num_rows
  Description: This batch file retrieves the top N classes from the database. You must provide exactly one numerical argument.
  - Example: all_cl.bat 3
- all_memberships.bat – Find a certain number of memberships from the database

class_name	schedule_id	class_id	staff_id	room_id	day_of_week	start_time	end_time	description	class_capacity
Spin	1	1	1	1	Monday	18:00	19:00	High-intensity indoor cycling session.	25
Yoga Basics	2	2	5	2	Tuesday	08:00	09:00	Relaxing beginner yoga flow.	20
HIIT	3	3	2	6	Wednesday	07:00	08:00	High Intensity Interval Training circuit.	20

  Usage: all_memberships.bat num_rows
  Description: This batch file lists the top N classes from the database.
  - Example: all_memberships.bat 10

membership_id	membership_type	duration_days	price
1	Basic Monthly	30	39.99
2	Premium Monthly	30	59.99
3	Annual Basic	365	399.99
4	Annual Premium	365	599.99
5	Student Monthly	30	29.99
6	Couples Monthly	30	69.99
7	Senior Monthly	30	34.99
8	Corporate Plan	30	49.99
9	Trial Week	7	9.99
10	Drop-In	1	5.0

- **memberships.bat –** Find members' active memberships

  Usage: memberships.bat first_name last_name
  Description: This batch file retrieves membership data for the specified member. You must provide exactly two string arguments.

- Example: memberships.bat Liam Johnson

member_id	first_name	last_name	membership_type	start_date	end_date
2	Liam	Johnson	Basic Monthly	2025-01-10	

- **usrenr.bat –** Find specific members enrolled in any class

  Usage: usrenr.bat FirstName LastName
  Description: This batch file allows a user to see the class(es) that he/she has enrolled in
  - Example: usrenr.bat Liam Johnson

Register for class: usrenr.bat /r FirstName LastName "Class Name" DayOfWeek StartTime
- Example: usrenr.bat /r Samuel Lenin "Yoga Basics" Wednesday 07:00

Cancel registration: usrenr.bat /cancel FirstName LastName "Class Name" DayOfWeek StartTime
- Example: usrenr.bat /cancel Samuel Lenin HIIT Wednesday 07:00
- **cl_info.bat**

first_name	last_name	class_name	room_id	day_of_week	start_time	end_time	status
Samuel	Lenin	HIIT	6	Wednesday	07:00	08:00	enrolled
Samuel	Lenin	Bootcamp	10	Wednesday	18:00	19:00	enrolled

Usage: cl_info.bat class_name
Description: This batch file retrieves info for any specific class you want.
- Example: cl_info.bat Bootcamp

class_name	schedule_id	class_id	staff_id	room_id	day_of_week	start_time	end_time
Bootcamp	27	10	10	37	Tuesday	07:00	08:00
Bootcamp	10	10	8	10	Wednesday	18:00	19:00

- **Room.bat –** Find the room the associated class is going to be held in
  Usage: pass one argument: room_id
  Description: This batch file lists a room that the class you registered for is happening in.
  - Example: room.bat 17

  Room name lookup:
  1. usrenr first_name last_name in terminal to find class you enrolled in along with the room_id

2. room.bat room_id to find the room name

- **Payments.bat –** View payments by member

  Usage: payments.bat first_name last_name
  Description: This batch file queries all payments for the specified member. You must provide exactly two string arguments.
  - Example: payments.bat Liam Johnson

9. Future Enhancements

• Integration with a web-based dashboard for analytics visualization.
• Terminal front-end framework for a visual appeal, user friendly, and feel more productized
• Support for mobile notifications about class availability.
• Pre-train an LLM to generate queries from text.