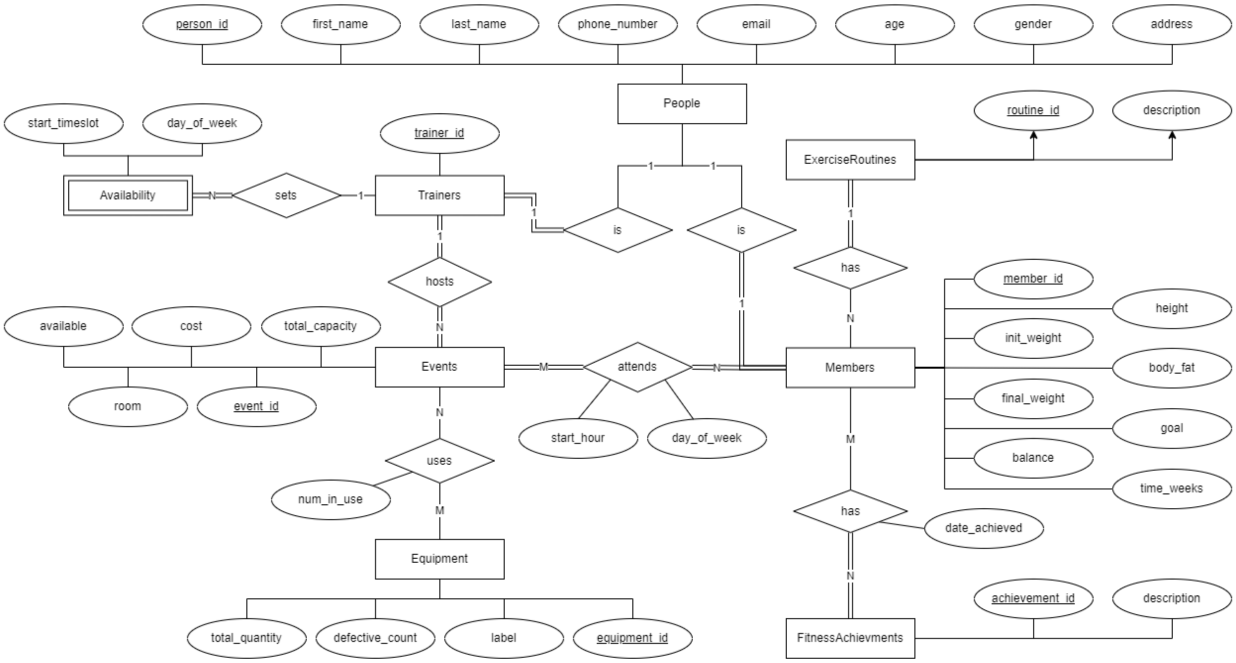
Health and Fitness Club Management System Report

Alvina Han, Michael Ge

Conceptual Design



Assumptions

- A member doesn't need to have a exercise routine or fitness achievements
- A member must attend at least one event
- Every event must be hosted by a trainer and attended by at least one member
- An event doesn't need to use an equipment and an equipment doesn't need to be used in an event

Figure 1: ER-Diagram

The Health and Fitness Club system is a database designed to capture the interrelations of several entities. The first entity, “People,” is an encapsulation of shared attributes of individuals that have relationships with “Trainers” and “Members” entities.

Members are distinguished with attributes relating to membership details and fitness goals. They are associated with “ExerciseRoutines”, where they lay out their routine, and “FitnessAchievements”, where they have their achievements, but they are optional to have. However, it is a requirement that members participate in at least one event.

Trainers have an associated “Availability” entity, which is a weak entity depicting their available time slots throughout the week. Trainers must hold at least one event as they are hired to be trainers at the club. Trainers don’t have an attribute exclusive to them; their significance is from their availability and the events they lead.

The “Events” entity represents scheduled activities within the club. It details attributes such as location, cost, and the number of members registered. An event needs to be hosted by one trainer, and at least one member must attend. This entity creates the interaction between the trainer and member(s). Also, events may utilize “Equipment,” though it is not compulsory.

“Equipment” entity has basic attributes of an equipment (total number, defective count) and they are not required to be part of an event, meaning that they may exist independently in the system.

The PNG to the ER-Diagram is in *Health-and-Fitness-Club-Management-System > Final Project Diagrams -> ER Diagram.png*.

Reduction to Relation Schema

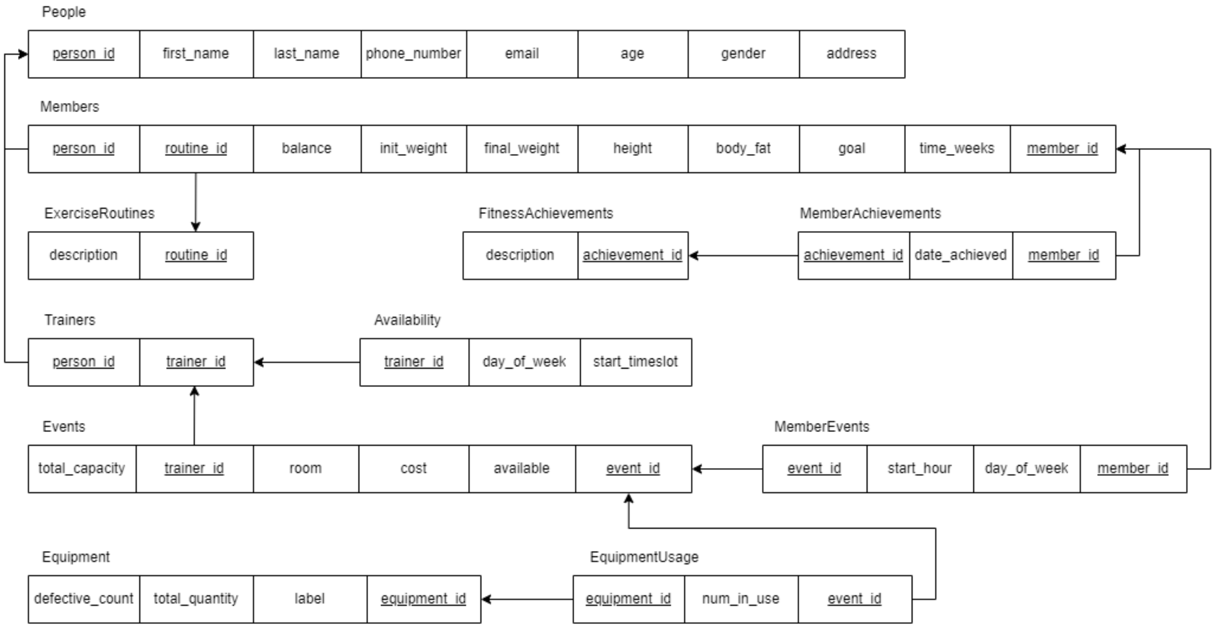


Figure 2: Relational Database Schema

The relational schema is designed based on the ER-Diagram (Figure 1).

The PNG to the Relational Database Schema is in *Health-and-Fitness-Club-Management-System > Final Project Diagrams -> Relational Database Schema Diagram.png*

DDL File

The Relational Database Schema(Figure 2) is transformed into a Data Definition Language statements file with a '.sql' extension. The file is in *Health-and-Fitness-Club-Management-System > SQL > ddl.sql*.

If encountered an error and want to restart, the clear.sql file will delete all the tables created by the DDL file. That file is in *Health-and-Fitness-Club-Management-System > SQL > clear.sql*.

DML File

A Data Manipulation Language file, with a '.sql' extension is provided with sample data for each table. The file is in *Health-and-Fitness-Club-Management-System > SQL > dml.sql*.

Implementation

The project is in the form of a Command-Line Interface and is programmed in Python.

Each Python script is dedicated to handling distinct aspects of the system:

dbConnection.py: this script it to establish connection to the database. It also executes passed in queries functions and handles errors if there are any.

Function Name	**Description**
createConnection()	Connects to PostgreSql Handles error
executeQuery()	Executes query (ie: insert, update table) - does not return Handles error
executeSelectQuery()	Executes query and returns result of the query Handles error

member.py: it handles member-specific operations. It enables user registration, manages member profiles, displays dashboard, and handles schedule.

Requirement	**Function Name**	**Description**
Profile Management	editProfile()	Edits member's personal (People table) and health (Members table) information
	viewProfile()	Views member's personal and health information *Global: function used in Member and Trainer

Dashboard Display	viewDashboard()	Displays member’s exercise routine, fitness achievements, and health information
Schedule Management	printSchedule()	Prinst events from the schedule that is passed in as an argument *Global: function used in Member and Admin
	viewSchedule()	Grabs member’s schedule and use printSchedule() to print
	addEvent()	Adds an event to member’s schedule
	removeEvent()	Removes an event from member’s schedule
N/A	memberSession()	Handles interaction with user who’s logged in as member and calls the correct function

trainer.py: it handles trainer-specific functionalities. It provides trainers with tools to manage their schedule and view member profiles.

Requirement	**Function Name**	**Description**
Schedule Management	setAvailability()	Set a trainer’s availability by specifying the day of the week, and start and end times
Member Profile Viewing	trainerSession()	View member’s profile (choose option=2)
N/A	trainerSession()	Handles interaction with user who’s logged in as trainer and calls the correct function

admin.py: it handles administration staff functionalities. It gives the user the privilege to book rooms, monitor equipment maintenance, update class schedules, and process billing and payment.

Requirement	**Function Name**	**Description**
Room Booking Management	roomAvailable()	Check if room is available at the specified time
Equipment Maintenance Monitoring	equipmentAvailable()	Check if changes in defective equipment will still allow for the events to take place
Class Schedule Updating	adminSession()	Update class schedule with the specified day of week and time (choose option=3)
Billing and Payment Processing	canPay()	Check to see if member has sufficient funds to pay
N/A	adminSession()	Handles interaction with user who’s logged in as admin and calls the correct function

customInput.py: a helper class that ensures user inputs are captured in the required format such as date inputs.

Function Name	Description
inputFormatted()	Take user's input
inputDateTime()	Take date input from the user (1) day of week (2) start and end time

main.py: the starting point of a program that executes from the command line. It integrates all the modules mentioned above and allow interaction with the user.

Requirement	Function Name	Description
User Registration	memberRegistration()	Registers members along with their personal and health information

GitHub Repository

GitHub - Health-and-Fitness-Club-Management-System

Video Demonstration is in ReadMe as well.