COMP 3005 B: Database Management Systems

Abdelghny Orogat

FINAL PROJECT REPORT

NIRAJ PATEL - 101269614

WINTER 2024

Abstract

This project involves designing and implementing a comprehensive Health and Fitness Club Management System for club members, trainers, and administrative staff. The application provides functionalities that allow members to register, manage their profiles, set personal fitness goals, and monitor their health metrics through a personalized dashboard.

Members can also reschedule or cancel personal training sessions and register for group fitness classes. Also, they can change their membership and leave feedback.

Trainers can manage their schedules and view member profiles, while administrative staff handle room bookings, fitness equipment maintenance, class schedules, and oversee billing and payments.

Member Functions:

1. Update Member Profile
2. Display Member Dashboard
3. Schedule Personal Training Session
4. Leave Feedback
5. Manage Membership

Trainer Functions:

1. Set Availability
2. View Member Profile

Admin Functions:

1. Manage Room Bookings
2. Update Equipment Status
3. Manage Class Schedules
4. Process Payments

The ER diagram mainly includes the following entities:

- **Members**: Individuals who use the club's facilities and services.
- **Trainers**: Professionals providing personal training services.
- **Admin Staff**: Employees responsible for administrative tasks.
- **MembershipTypes**: Different types of memberships are available for purchase.
- **PersonalTrainingSessions**: Sessions booked by members with specific trainers.
- **Classes**: Fitness classes are scheduled so that multiple members can join.
- **RoomBookings**: Booking details of rooms for various activities.
- **Equipment**: Inventory of fitness equipment available at the club.
- **Payments**: Transactions related to the services availed by the members.
- **Class Registrations**: allows members to join group classes.
- **Feedback**: Members can submit reviews/rate sessions
- **Trainer Availability**: Trainers can set their availability.

****NOTE: ER Diagram and Schema Diagram are also attached in the repo. Follow ReadMe for more instructions.** (in case the below images are not clear)

Relationships:

- Members are central, connecting to PersonalTrainingSessions, ClassRegistrations, and Payments.
- Trainers link to PersonalTrainingSessions, Classes, and TrainerAvailability.
- Classes connect through ClassRegistrations to Members and are taught by Trainers. Payments tie Members to MembershipTypes.
- Feedback involves Members, Classes, and Trainers.
- User Authentication ties user accounts to Members, Trainers, or Admin Staff depending on the role.
- Membership Type Audit tracks changes in Membership Types.

Cardinalities and Participation Types:

- A **member** can book multiple personal training sessions and group classes but at least one of either. Each session or class is related to exactly one member.
- **Trainers** can conduct multiple personal training sessions and group classes, mandatory participation in at least one session or class.
- Each **payment** transaction is linked to one member, indicating mandatory participation of members in payments if services are used.

The following assumptions were used for this database creation;

1. Each **member** can have multiple training sessions but only one trainer manages each session.

2. **Trainers** can manage their schedules directly and are assigned to both personal training sessions and group classes.

3. **Payments** are linked directly to members and the services availed.

4. **Equipment maintenance** is scheduled and tracked through the system.

5. **Unique Email Addresses**: Every member, trainer, and admin staff has a unique email -> individual identification and authentication.

6. **JSON Data Format**: Health metrics and fitness goals are stored in a structured JSON format -> systematic data handling/ retrieval capabilities.

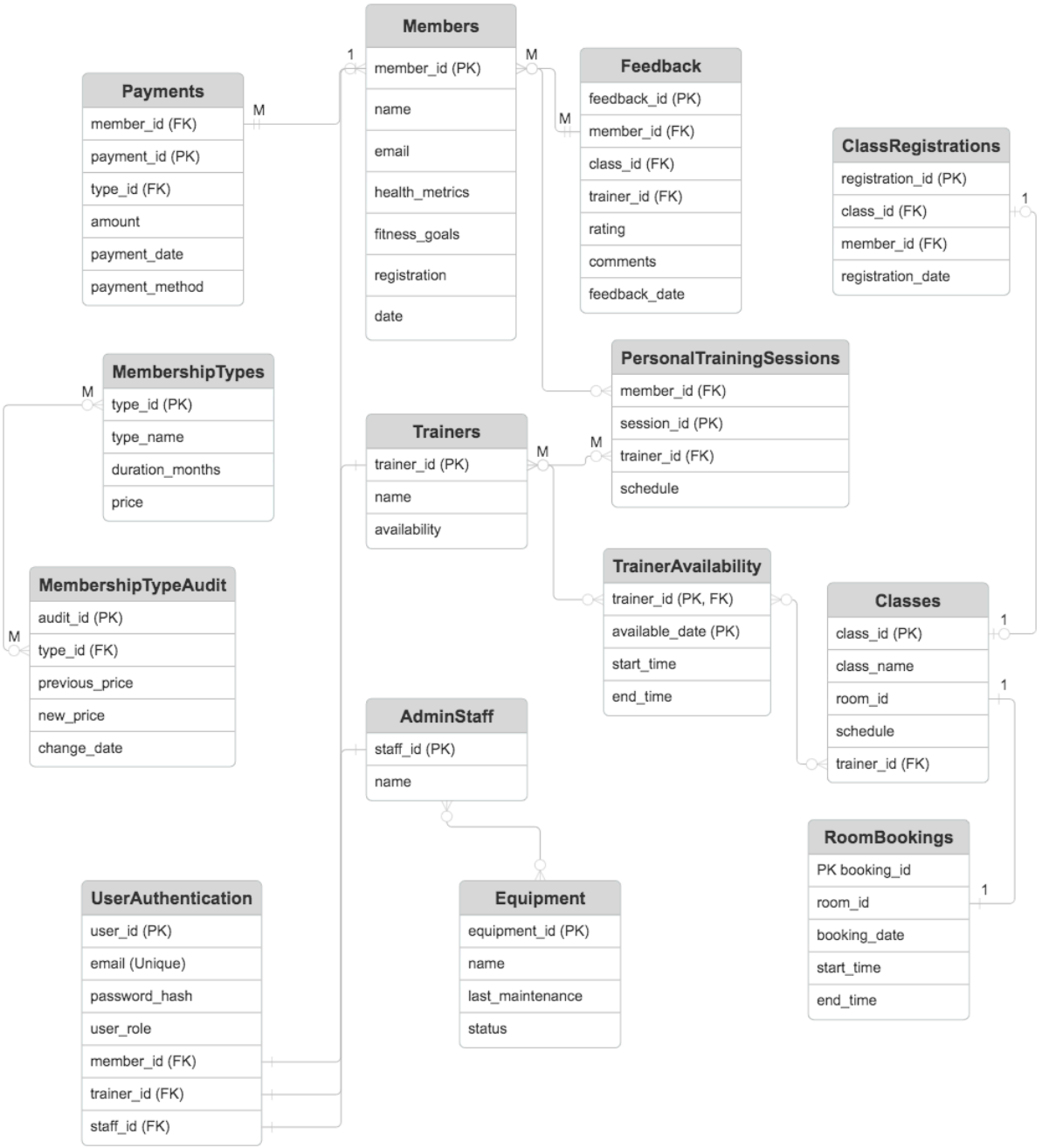
7. **User Roles**: Distinct roles (members/trainers/admin) are defined

8. **Trainer Scheduling**: A dedicated table for trainers' availability to avoid conflicts in time, and double-boking issues.

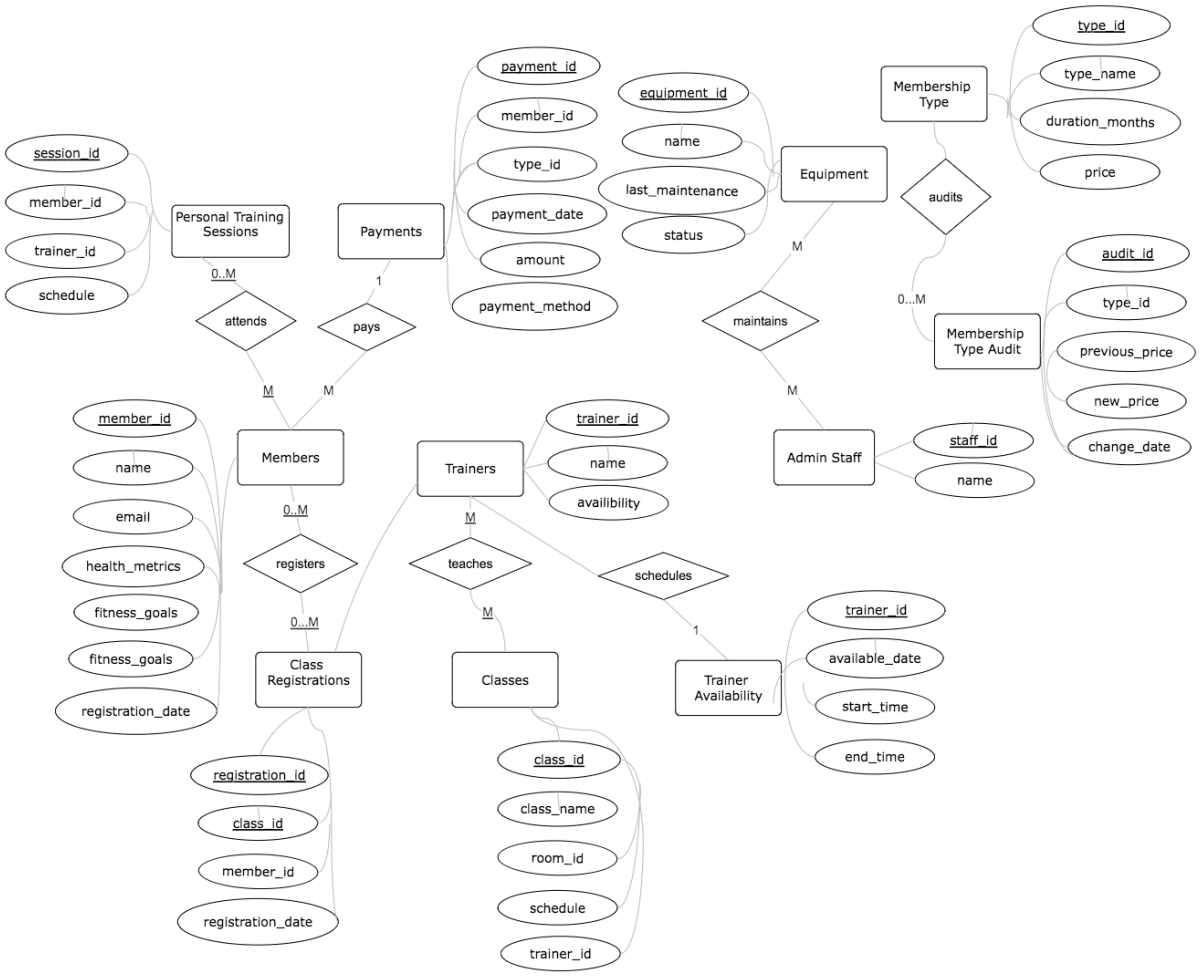
9. **Membership Management**: The system tracks different types of memberships and any changes in membership details -> auditing purposes.

10. **Feedback Mechanism:** Members can provide feedback on both classes and trainers.

Schema Diagram



ER Diagram



****NOTE: Please refer to ReadMe for DDL and DML files. (attached separately in the repo).**

Implementation:

The application is implemented as a **Command-Line Interface (CLI) using Python.** It interacts with a **PostgreSQL database**, utilizing **psycopg2** for database connectivity.

Modular architecture -> separate modules for handling different functionalities such as user interactions, database operations, and specific tasks for members, trainers, and administrative staff.

Bonus Features Implemented:

1. **Indexes** -> frequently accessed fields such as **Member_ID** and **Trainer_ID** to speed up query performance => operations involving member profile lookups and session scheduling.

2. **Triggers** ->

    a. **The trigger for Logging Changes in Membership Type Prices:**

This trigger is designed to **log any changes in the membership type prices**. Whenever there is an update to the price in the MembershipTypes table that alters the current price, the change is recorded -> auditing purposes and tracking price changes over time.

Activation: Before an update on the MembershipTypes table.

    b. **Trigger to Validate Email Format:**

This trigger ensures that the **email addresses entered** in the **UserAuthentication** table are in a **proper email format**. This validation helps prevent data corruption and improves data integrity by ensuring that all email addresses are in a correct and usable format.

Activation: Before an insert or update on the UserAuthentication table.