ANJISH PRADHANANGA | 101260233

2 Project Report

2.1 Conceptual Design

The conceptual design of the database for the Health and Fitness Club involves identifying the entities and relationships within the system. Here's an overview of the ER-diagram:

Entities:

Users: Represents all individuals registered in the system. Includes attributes such as user ID, email, first name, last name, password, date of birth, city, country, phone number, and state.

Members: Represents members of the fitness club. Inherits general user attributes from the Users entity and includes additional attributes like height and weight specific for members.

Trainers: Represents trainers employed by the fitness club. Inherits attributes from the Users entity and includes a specialty field specific for trainers.

Admins: Represents administrators with special privileges within the system. No specific attributes added for now but can be added according to use-case.

Classes: Represents fitness classes offered by the club. Includes attributes such as class ID, name, description, type, trainer ID, and price.

Rooms: Represents physical rooms within the club where classes take place and their availability status.

Equipments: Represents equipment available in the fitness club and their functioning status.

Schedules: Represents scheduled classes. Includes attributes like schedule ID, date, start time, duration, class ID, room ID, and status. Schedules are uniquely categorized by their date, time and duration. The conflicts in scheduling are automatically handled by the system.

Fitness Goals: Represents fitness goals set by members. Includes attributes like goal ID, member ID, title, description, value, start date, target date, achieved date, and status.

Invoices: Represents invoices generated for members. Includes attributes like invoice ID, member ID, amount, date, schedule ID, and status. Uniquely identified by the scheduleID and memberID.

Exercises: Represents exercises available for members to perform. Includes attributes like exercise ID, name, description, and type.

Member Exercises: Represents exercises performed by members. Includes attributes like member ID, exercise ID, reps, weight, and start week.

Relationships:

One-to-One:

Users to Members (A user can be a member).

Users to Trainers (A user can be a trainer).

Users to Admins (A user can be an admin).

One-to-Many:

Members to Fitness Goals (A member can have multiple fitness goals).

Members to Invoices (A member can have multiple invoices).

Classes to Schedules (A class can have multiple schedules).

Trainers to Trainer Availability (A trainer can have multiple availability slots).

Members to Member Exercises (A member can perform multiple exercises).

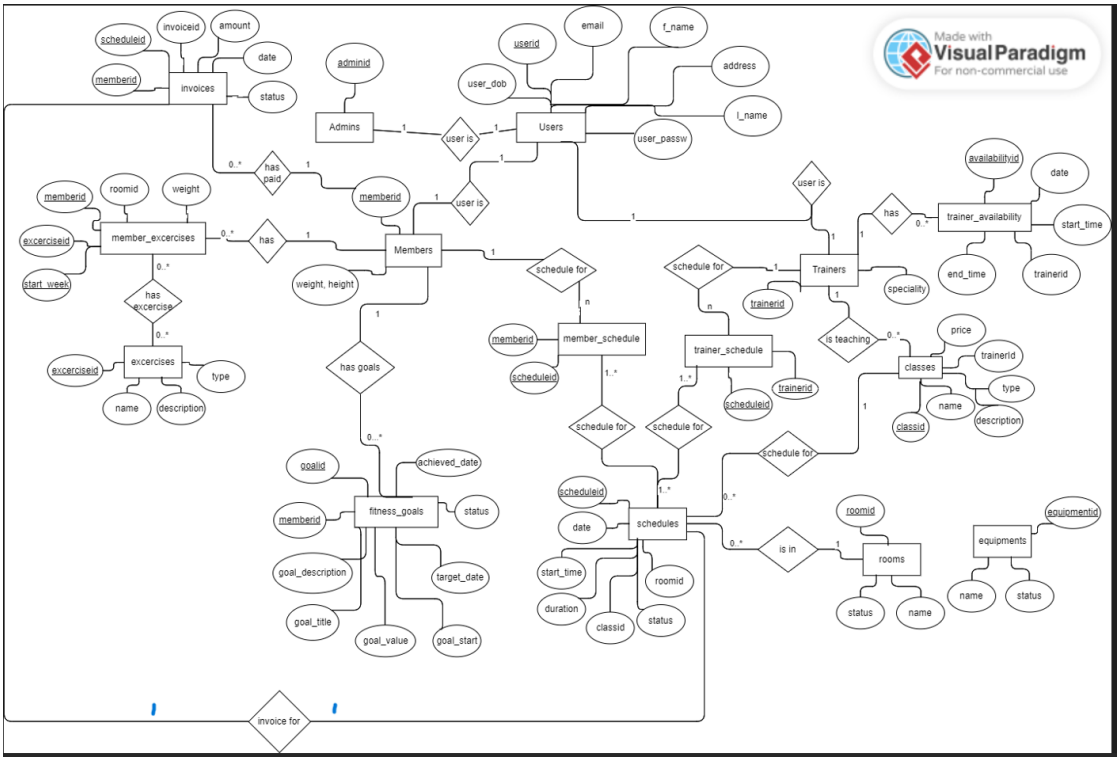
Trainers to Schedules (A trainer can conduct multiple schedules, and a schedule can have one).

Many-to-Many:

Members to Schedules (A member can attend multiple schedules, and a schedule can have multiple members).

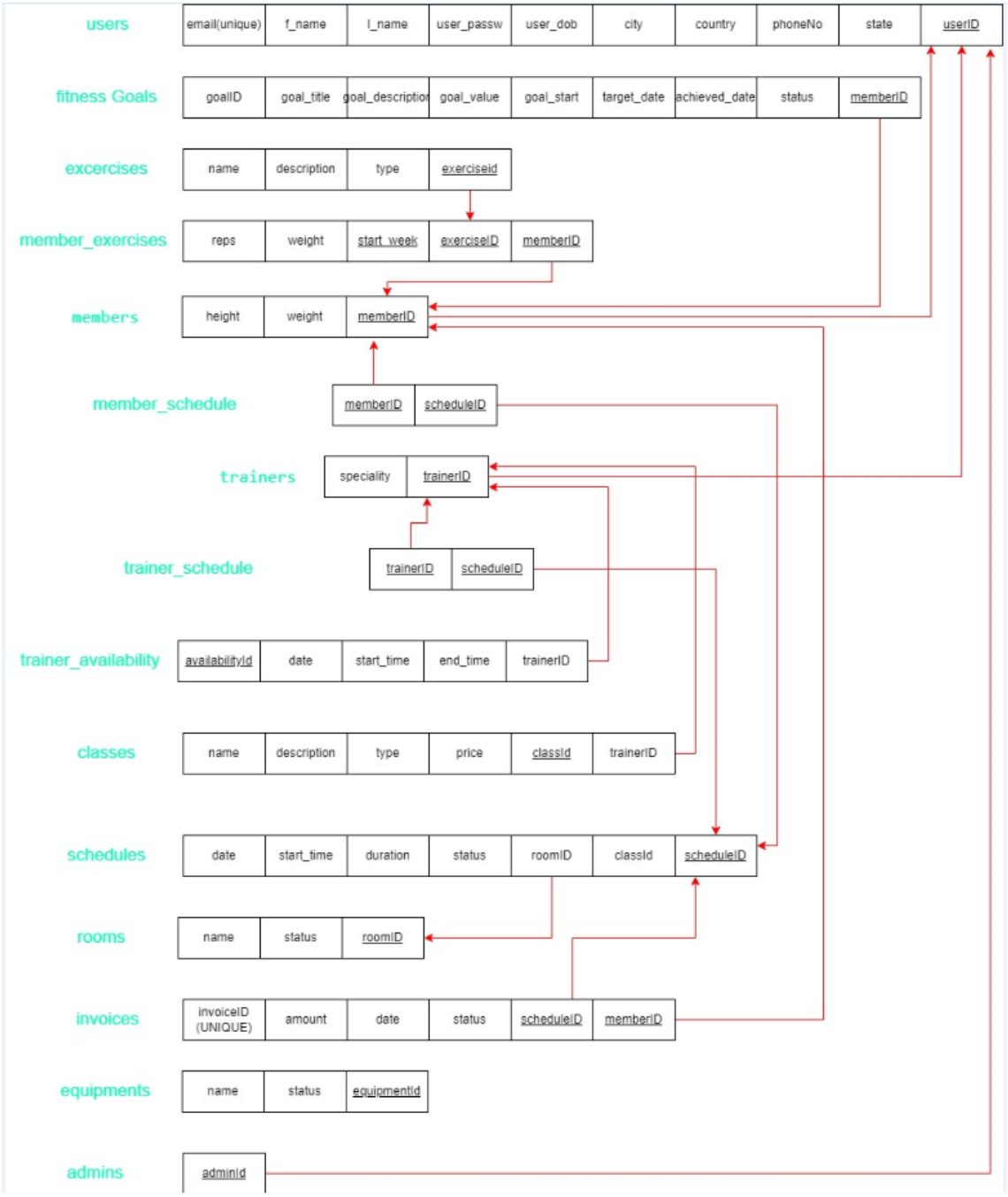
2.2 Reduction to Relation Schemas

ER Diagram:
Check ` /ER.vpd.png `



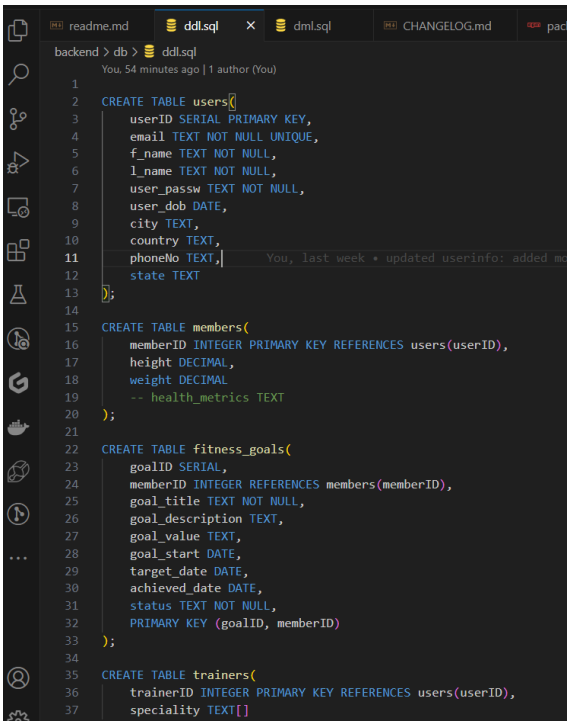
Relational Schema Diagram:

Check ./relational_schema.jpg


users
email(unique)
f_name
l_name
user_passw
user_dob
city
country
phoneNo
state
userID
fitness Goals
goalID
goal_title
goal_description
goal_value
goal_start
target_date
achieved_date
status
memberID
excercises
name
description
type
exerciseid
member_exercises
reps
weight
start_week
exerciseID
memberID
members
height
weight
memberID
member_schedule
memberID
scheduleID
trainers
speciality
trainerID
trainer_schedule
trainerID
scheduleID
trainer_availability
availabilityId
date
start_time
end_time
trainerID
classes
name
description
type
price
classId
trainerID
schedules
date
start_time
duration
status
roomID
classId
scheduleID
rooms
name
status
roomID
invoices
invoiceID (UNIQUE)
amount
date
status
scheduleID
memberID
equipments
name
status
equipmentId
admins
adminId

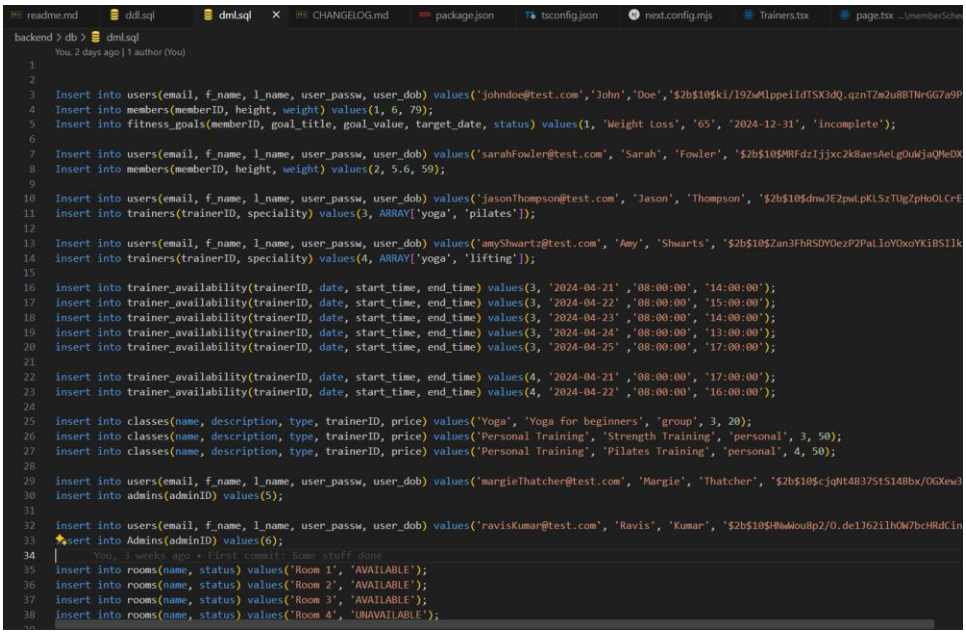
2.3 DDL File

Check ‘./backend/db/ddl.sql’



2.4 DML File

Check ‘./backend/db/dml.sql’



2.5 Implementation

The Implementation is a Web-Application which consists of two ends:

- Backend
    - Implemented in Node.js and Express.js.
    - Used to host APIs to communicate with backend.
    - Drizzle ORM for 'node-postgres' used to facilitate querying with the PostgreSQL database.
    - Authentication done using JWT.
    - Password hashing done using ` bcrypt` library.
    - Refer to `readme.md` file in './backend' to see the API documentation.
- Frontend
    - Implemented using TypeScript and Next.js framework.
    - Used by the end user to access and update data.
    - Material-Ui used for styling.

2.6 Bonus Features (Optional)

- A full-scale web-Application is made using Next.js for frontend and APIS for backend using Express.js.
- The User interface is aesthetically pleasing, well laid out, intuitive and mobile-responsive. Icons, interactive modals and cards are used to display information well.
- User authentication is done on both client-side and backend for more robustness and security using JWT authentication. Invalidated/tampered/expired tokens/users cannot access secured data by calling the APIs.
- Users can sign-up with multiple roles and access to specific content is automatically verified by the backend using an auth middleware according to the registered role.
- Sensitive user information like passwords are stored in the database after being encrypted.
- Data validation and error handling/logging is done on both frontend*(99%)* and backend.
- Done by a one-man team!

2.7 GitHub Repository

Github repo:

https://github.com/Titan0932/FitBro/tree/main

I know it says put the ddl and dml files in directory called 'sql' but I put it in './backend/db' .
Hopefully this is not an issue!

Video Demonstration:

Btw, there are a couple things that are only slightly different from the specs. Not necessarily too different but maybe a little more complicated for practical reasons as I thought they made more sense. But essentially, the same updates are being done on the database even though via a different path.

There's a lot of things to show and I will try my best to wrap it up in just 15 minutes! I will be speeding up the video! Please do send me an email if anything's unclear or any clarifications needed!!

Link: https://youtu.be/L9qeq7prMrc