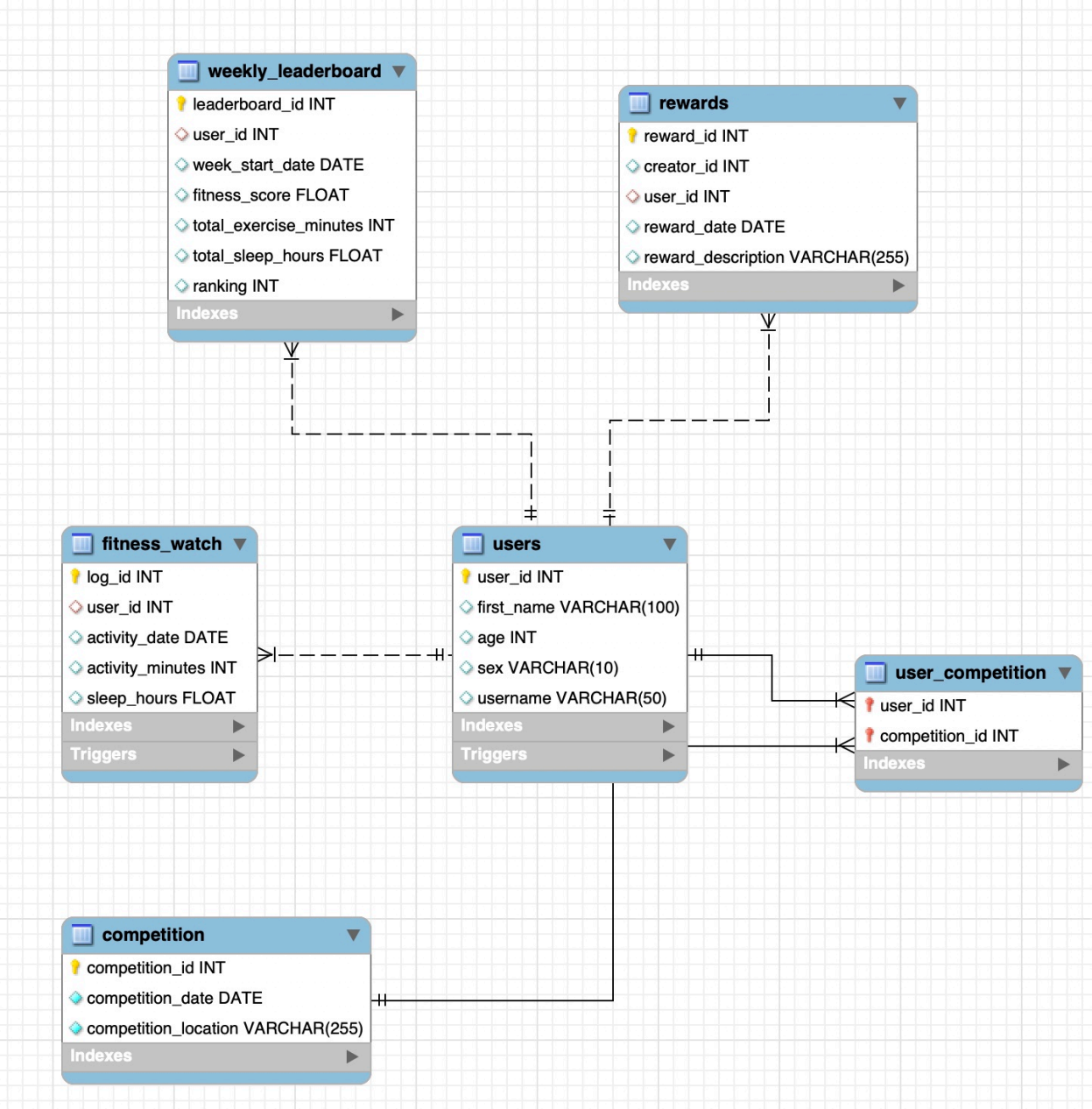
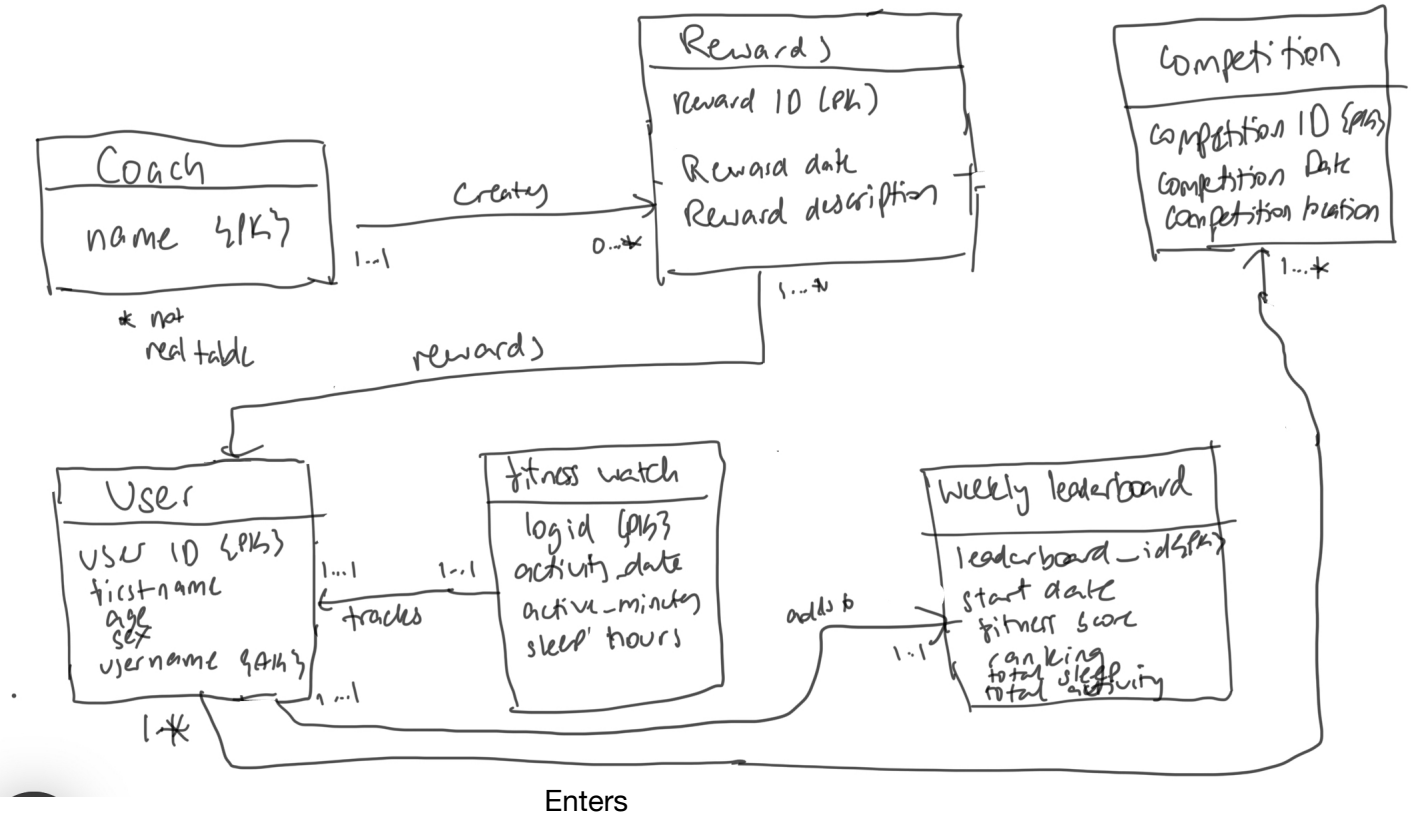
Neil Israni

Coach Buddy





You will be given a menu like this:

Select an option:

1. Create User
2. Create Fitness Log
3. Create Rewards for the Week
4. Create Competition
5. View Weekly Leaderboard
6. View Rewards for a User
7. View Personal Report for a User
8. View Competitors
9. Update User
10. Update Activity
11. Check Competition Eligibility
12. Register User for Competition
13. Delete Activity
14. Delete User
15. Delete Reward
16. Exit

Building Coach Buddy has significantly enhanced my technical expertise in developing a responsive database system and integrating it with a Python backend. I learned to efficiently use MySQL for managing structured data and to create seamless interactions between Python and MySQL. Managing both the backend and database independently required meticulous planning. Separating these responsibilities allowed me to focus on each component without overwhelming overlap, but it also highlighted the need for more efficient time division. For example, while the backend took precedence initially, integrating and testing the database later in the process required backtracking, which could have been avoided with a parallel workflow.

Designing tables for user activity and reward systems gave me valuable insights into modeling fitness data and understanding how such systems reflect user engagement. The challenge was ensuring a logical structure that supported extensibility—for example, linking fitness logs to a leaderboard while also allowing competition eligibility checks based on dynamic reward points.

I considered building a more visually appealing frontend, possibly with Next.js, to replace the command-line interface. However, I'm glad I prioritized backend functionality over frontend development in this version. A heavy focus on frontend at this stage might have diverted attention from essential features like fitness reporting and competition eligibility, which form the project's core.

I encountered issues database speed. Fetching reports or updating fitness logs can sometimes be slow, requiring multiple attempts. I request the grader to also not try to delete tuples hardcoded in, and experiment by creating and deleting new ones.

Future Work

- **Developing More Fitness Metrics and Insights:**
  Incorporating advanced statistics and visualizations (e.g., weekly progress graphs, activity trends) to provide users with actionable insights into their fitness journeys.

- **Integrating NLP for Personalized Recommendations:**
  Using frameworks like **Hugging Face**, I can add conversational insights, such as offering personalized activity suggestions or summaries of weekly performance, making the system more user-friendly.

- **Improved Competition Features:**
  Expanding competition functionality to include automatic pairing of users for challenges or group fitness events.

- **Gamification:**
  Adding gamification features like badges, streaks, or achievements to drive user engagement.

- **Mobile App Compatibility:**
  Building a mobile-friendly frontend to make fitness logs and reports easily accessible on the go.

Moving forward, I plan to refine this foundation, optimizing performance and adding user-friendly enhancements to create a tool that not only tracks fitness but also actively motivates and guides users toward their goals so socially.