Database Schema modelling and Entity Relationship Diagram (ERD) for an Online Multiplayer Chess System

By Elochukwu (Gerald) Okafor
FlexiSAF Internship Backend (Advanced) Track Task #2

Introduction

As part of our advanced backend internship program, the second assignment required us to design a complete database schema and corresponding ER diagram for a domain of our choosing. I selected a real-time online multiplayer chess platform as my domain, focusing on the core entities and relationships needed to support player management, game sessions, and move tracking.

Identifying Entities

The first step was identifying the core entities to model the database around. For a chess system similar to chess.com, the most important entities are:

- **Players**: User accounts and profiles
- **Games**: Individual chess matches between players
- **Board States**: Snapshots of the board at different points in time
- **Moves & Move History**:Individual moves and complete game records

Database Schema Modelling

I used an online tool: dbdiagrams.io, to model the entities and their relationships with relevant constraints.

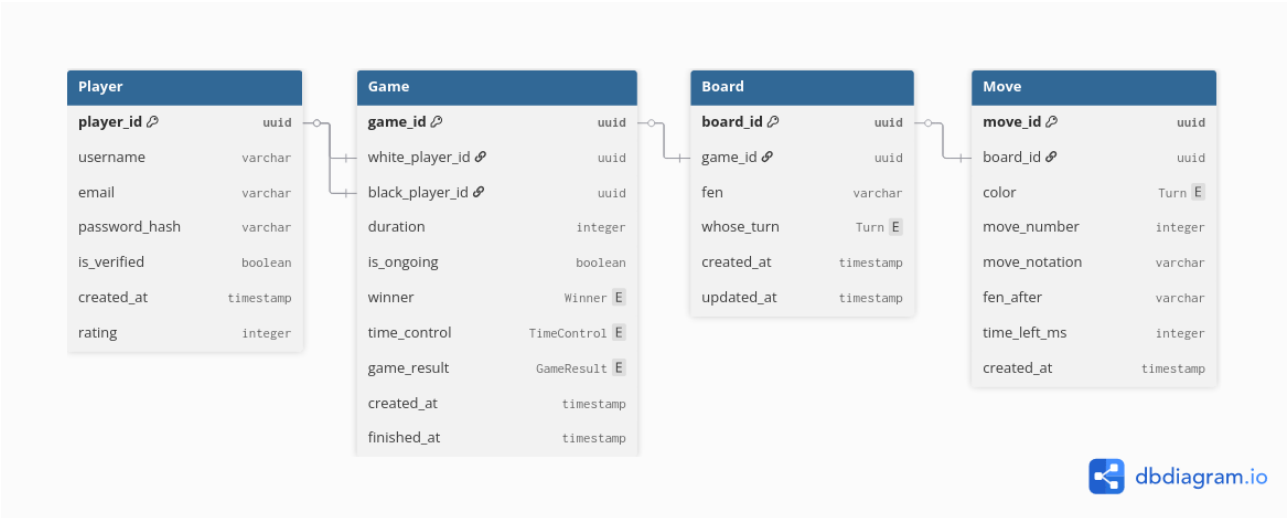

Figure 1: Database Schema Design

The Player entity consists of the typical username, email address, and password, along with an ELO rating that defaults to 1200 and other metadata.

The Game entity includes references to two players (White and Black) and contains other game-relevant information, such as winner, duration, and time control.

A Chess Game is defined by the Board State, which is another entity. It contains whose turn it is to play, the FEN game string, and a reference to a game.

Finally, there are Move entities which compose a Move History, it captures a snapshot of the board state, including details such as time left, and the board FEN string after the move is made.

Entity-Relationship Diagrams

I used another web tool: draw.io to design the ER diagram according to the schema I modeled above. The diagram illustrates the relationships between all four core entities and their cardinalities.

Each Player can participate in multiple Games (N:M relationship), while each Game contains multiple Board States representing different positions throughout the match (1:M relationship).

The Move entity captures individual player actions and links to specific Board States, creating a complete audit trail of game progression.

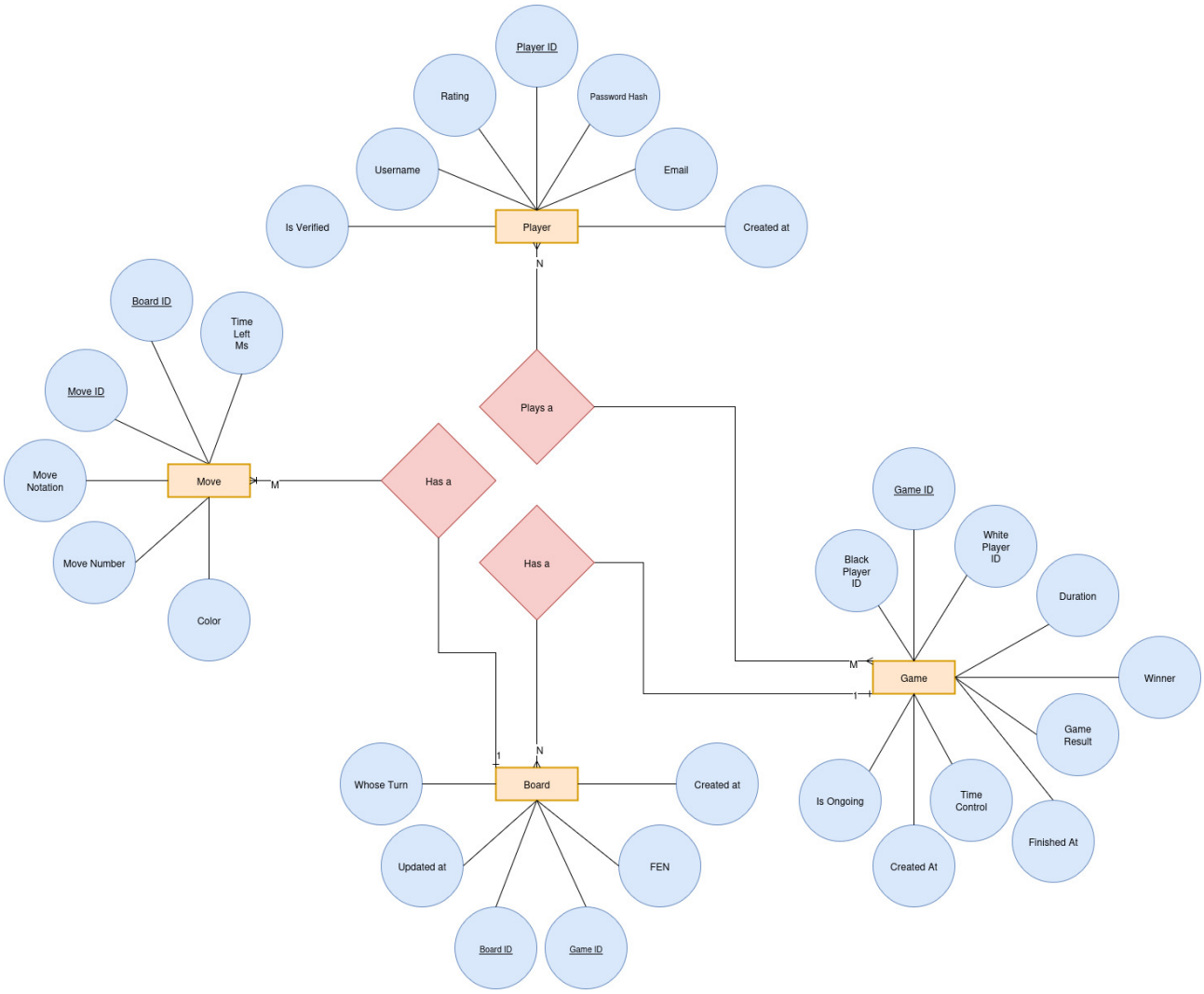

Figure 2: Entity-Relationship Diagram