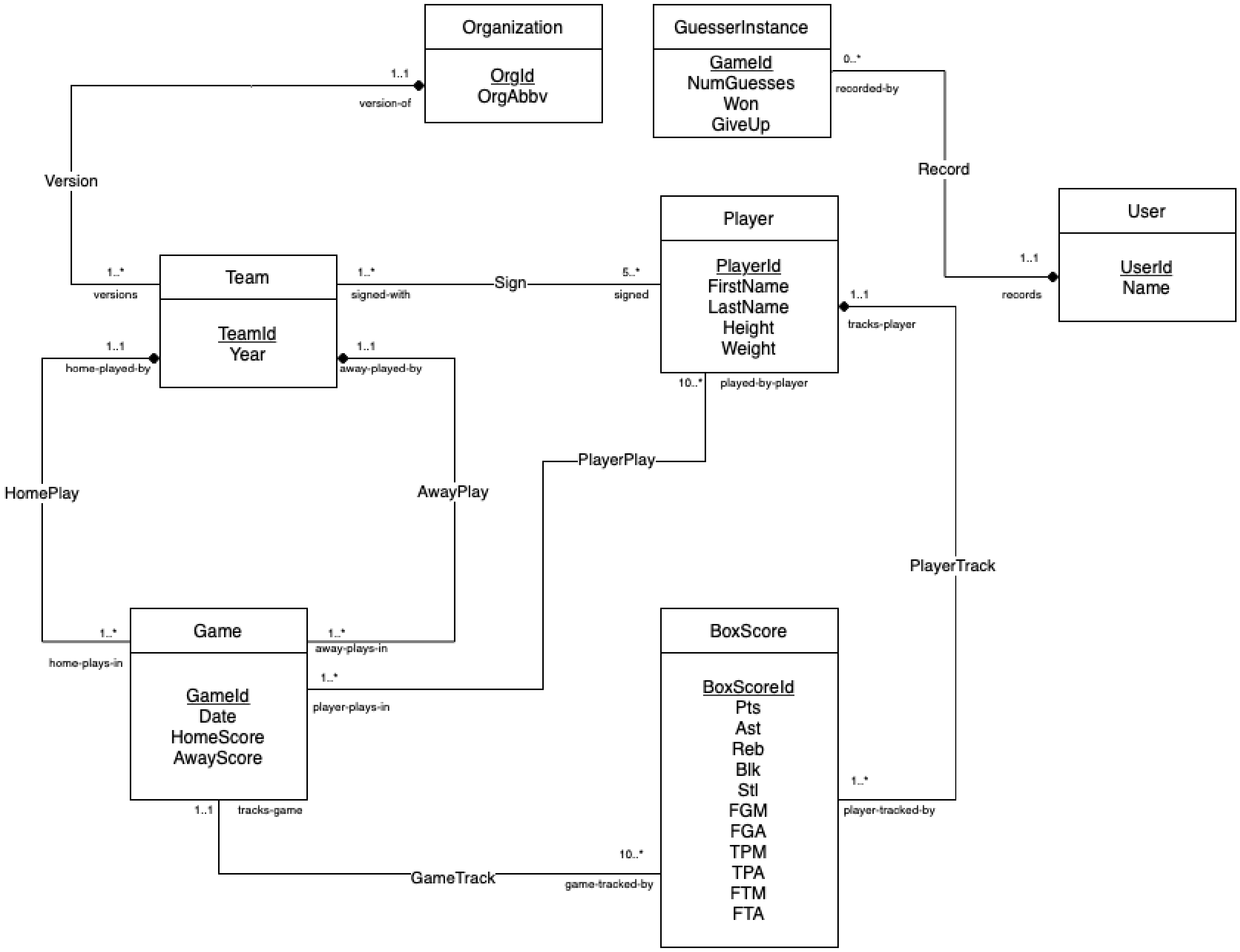
Q-team107-teamBlue
CS 411 Section Q

Conceptual and Logical Database Design

UML Diagram:



Note:
The GuesserInstance and User entities are stretch goals that we'd like to implement if time permits.

Entity Descriptions:

- Organization: An organization in the NBA, such as the Golden State Warriors.
- Team: A team in the NBA of a given year, such as the 2022-2023 Golden State Warriors.
- Game: An NBA game in which 2 teams play each other.
- Player: An NBA player with physical attributes.
- BoxScore: The individual statistics of one player during one game.
- GuesserInstance: A single play of our guessing game, which has corresponding information and statistics about that game.

- User: A user who plays our guessing game.

Multiplicity Assumptions and Relationship Descriptions:

- **Organization** and **Team:**
    - We assume that an organization has at least 1 instance of a team. For example, the **organization** *Golden State Warriors*, has at least one **team** such as the *2022-2023 Golden State Warriors*
    - Conversely, a team should only have one organization. A team that played in a certain season can not have zero and two different organizations that they played under.
- **Team** and **Player:**
    - We assume that a team has signed at least 5 players and a player has signed with at least 1 team in their career. For example, the **team** *2021-2022 Chicago Bulls* has at least five players signed to the team that year such as *Demar Derozan, Ayo Dosunmu, Zach Lavine, Alex Caruso, Lonzo Ball*, etc. and the **player** *James Harden* can be signed to at least one team, being the *Philadelphia 76ers* and *Brooklyn Nets*
- **Game** and **Team**
    - A **Game** can have only 1 **Home Team** and 1 **Away Team,** requiring 2 teams total. Therefore the multiplicity is 1..1 for each team.
    - A **Team** can play multiple games in a season (specifically 82 in the regular season), and we assume that a team must play at least one game in a season. Therefore, the multiplicity is 1..*.
- **Player** and **Game**
    - A **Player** must have played in at least 1 **Game** and a **Game** must be played by at least 10 players.
    - This is because in the NBA, each game requires 2 teams and with at least 5 players playing on each team → requiring at least 10 players in a **Game**
    - The game requires that a **Player** has played in the NBA and so, a player must have played at least 1 **Game.**
- **Player** and **BoxScore**
    - A **BoxScore** can only have one player since each **BoxScore** can only refer to one player at a time. Therefore they have a 1..1 multiplicity.
    - Conversely, A player can have multiple **BoxScore**s since each game they played will be tracked by the BoxScore. We assume that a player must have at least 1 BoxScore, or they would not be considered players. Therefore they have a 1..* multiplicity.
- **User** and **GuesserInstance**
    - A **User** is defined by a userId, and can have 0 to many **GuesserInstances** since a player can play multiple games.

    - A **GuesserInstances** can have exactly 1 **User** since **User.userId**, will be a foreign key to **GuesserInstances**. Multiple players cannot play an instance of a game. This is a single player game.
- **Game** and **BoxScore**
    - A **Game** must have at least 10 **BoxScores** because a game must have 10 players and each player has a **BoxScore**
    - A **BoxScore** must only have 1 **Game** because a **BoxScore** records each game's stats

We plan to enforce some of the x..* constraints where x > 0 by implementing triggers. However, since triggers can be time inefficient (especially with large databases), and many of these constraints are not vital to the playability or logic of our game, we may choose to simplify the assumption to 0..*

SQL Relational Schema

- Organization(OrgId: INT [PK],OrgAbbv: VARCHAR(225))
- Team (TeamId: INT [PK], Year: INT, OrgId: INT [FK to Organization.OrgId])
- Player(PlayerId: INT [PK], FirstName: VARCHAR(225), LastName: VARCHAR(225), Height: VARCHAR(225), Weight: INT)
- Game(GameId: INT [PK], Date: DATE, HomeScore: INT, AwayScore: INT, HomeTeamId: INT [FK to Team.TeamId], AwayTeamId: INT [FK to Team.TeamId])
- BoxScore(BoxScoreId: INT[PK], Pts: INT, Asts: INT, Rebs: INT, Blks: INT, Stls: INT, TOs: INT, FGM: INT, FGA: INT, TPM: INT, TPA: INT, FTM: INT, FTA: INT, PlayerId: INT [FK to Player.PlayerId)
- Sign(TeamId: INT[PK], PlayerId: INT[PK])
- PlayerPlay(GameId: INT[PK], PlayerId: INT[PK])
- GuesserInstance(GameId:INT[PK], NumGuesses:INT, Name:VARCHAR(255), GaveUp: BOOLEAN, UserId:INT[FK to User.UserId])
- User(UserId: INT [PK], Name:VARCHAR(255))

Cardinality:

- Organization
    - We assume that there will be about 30 organizations. Currently in the NBA there are 30 organizations, and although they have undergone changes with regards to their name, they are considered one organization. For example, the Seattle Supersonics and Oklahoma City Thunder, although different teams, were under the same organization.
- Team
    - We assume that there are 150 instances of teams. Since our game is pulling from data of the last 5 years in the NBA, there would be 30 instances of Team for each season, for 5 seasons.

- Player
    - Since it is difficult to approximate the number of players that play a NBA game within a 5 span, for the sake of estimating the cardinality of the table, we assume that all players between teams and seasons are unique. This would be the upper bound of the table: 15 x 30 x 5 = 2,250 players.
- Game
    - There are 1,230 games in a season in the NBA, and with a 5 season span, the cardinality of this table will be approximately 6,150.
- BoxScore
    - There are 6150 games in 5 years per above and 30 players play in each game, making the cardinality 184,500.
- Sign
    - Each of the 30 teams will have approximately 15 players, making the cardinality 30 x 15 = 450.
- PlayerPlay
    - From above, we assume that there are 2,250 total players. Assuming each player plays 82 games, the upper bound for this table is 2,250 x 82 = 184,500
- GuesserInstance
    - The cardinality depends on how popular our game becomes and the enjoyment of each User that they play more than one game. (hopefully it's widely used!)
- User
    - Same as for GuesserInstance, the cardinality depends on how popular our game becomes.

We recognize the fact that the tables are significantly large, specifically BoxScore and PlayerPlay. If faced with time efficiency or budgetary constraints, we may scale down the dataset from the last 5 seasons to 1-2 seasons. If our game contains two seasons, we will still include retired players which was one aspect of our game that separated us from similar games. Otherwise, if we only include one season, our game still differs drastically by including team performance, player aggregate performance, player performance in individual games, and player physical attributes.