Details For Relationships, Participation And Cardinality Constraints

Coach - Has - CoachStats

- Each coach in `coaches_24_25.csv` **has** a of their stats based on Current(This season), Franchise(seasons with the current team), and Overall(career totals in the NBA). Coaches in the NBA will have these game records displayed after seasons.

- The has relationship is one-to-many since each coach will have a playoff and normal record. This is because all NBA games they have coached are thought of seperately in our database of all time wins, losses and games played. A game record has 1 coach that it is tied to since we do not keep track of a record for multiple coaches that would logically not make any sense.

- Each Coach only has to partially participate in this relationship since a coach can exist in the database without the CoachStats. The CoachStats must totally participate since the CoachStats would not make sense if there was no coach assigned to that CoachStats. This is also why CoachStats is a **Weak entity**.

- The Primary key in this relationship is CoachID. This will be sufficent of uniquely identify the entities since CoachStats is a weak entity.

Coach - Instructs - Team

- The reason for this relationship is each coach will instruct a team in an NBA season. Since we got coaches and teams for 2024-2025 this relationship will focus on that particular season.

- The instructs relationship is one-to-many. Each coach can instruct one team. This is because in our `coaches_24_25.csv` we have a coach only instructing one team in the dataset. Each team is instructed by multiple coaches in a season as well since coaches can swap out with each other for games in the season.

- Each coach will totally participate in this relationship since in `coaches_24_25.csv` has an associated team along with their stats in a season. There is no coach without a team in that file. A team must totally participate as well in the instructs relationship since each team in the season is instructed by at least one coach.

- The primary key in the instructs relationship is (CoachID). This will uniquely identify every entity in the instructs relationship from one another.

Team - Play - Game

- For this relationship we will have the `team.csv` and `game.csv` connect from their IDs. Teams play in NBA games or games would not make any sense. Thus the reason this relationship exists.

- This relationship is going to be a many-to-many relationship since each team in the NBA does play many games and each game is played by many teams which it is actually 2 games no more or less. As you can see by the `Game.csv`.

- Game in the play relationship totally participates since each game needs to have 2 teams play for the game to occur or exist in the database. Teams in the play relationship only need to partially participate since a team can still exist without playing a game.

- The primary key in the play relationship will be: (TeamID and GameID). This will be sufficent to clearly identify any entity unqiuely.

Team - HasHome - Arena

- In the newly added dataset `Arena.csv` will connect a team having a home arena by their TeamID. In NBA a team can have a specific home arena and the dataset does reflect this.

- The owns relationship is one-to-one this is because each team will own an arena in NBA and every arena is owned by one team. As we can see in `Arena.csv` this is the case.

- In the owns relationship the team has partial participation since a team can exist in the database without having an assigned arena. They would be able to play games and always be on visitor teams. Arenas have to totally participate in this relationship since we would not have arenas that dont belong to an NBA team in our database.

- The primary key for this owns relationship will be either (TeamID or Arena name) it is safe to assume that there will never be a TeamID identical to another one in the NBA and an arena name will never be named the same as another in the NBA as well.

Team - Drafts - Player

- This relationship exists since a team will draft a player to play on their team in the NBA. We have connectedness from the `drafts_history.csv` with teamID and PlayerID that connect the entities together.

- The Drafts relationship is 1-to-many since each player will be drafted once in the NBA playing for different teams and a team will draft many players overtime in seasons to play for their team. We can see that is the case in `drafts_history.csv`a player can be drafted olny once.

- The Drafts relationship are both partially participate in this relationship since a player does not need to be drafted on a team to be in our database. We have undrafted players in `Player.csv`. A team does not need to have players on it to exist in the database.

- The primary key in this relationship will be (Player_ID) to uniquely identify every entity in the database.

Player - Plays in - Game

- In the NBA players need to play in games for the game to even be considered a game. `Play in.csv` will have that information of players playing in a specific NBA game

- The plays in relationship is many-to-many since each player will play many games in a season and each game will have many players in the game.

- In the plays in relationship a player will partially participate since a player does not need to play in a game to be considered in a database. Each game will totally participate in this relationship since games need players to play in the game or the game would not be considered in the database and they would have the game as a write off.

- The primary key in this relationship will be (PlayerID and GameID). This will be enough to uniquely identify every player stat since each players stats are only considered once in every game.

Arena - Hosts - Game

- In the NBA a game needs to be hosted at an arena for the game to be played. Thus the need of this relationship. `Game.csv` will have which arena the game is hosted at.

- The Hosts relationship is one-to-many. Each game is hosted by one arena. This is because a single game cant be in many arenas. Each arena hosts many games. Many games in a season will have one arena host many games in it.

- In the Hosts relationship a Arena will partially participate since an arena can exist in the database that has no games played at it and a game totally participates in the relationship since a game has to be played somewhere and will not exist without an arena to play at.

- The primary key in this relationship will be (Arena Name and GameID) this is sufficent since this will uniquely identify every entity

Player - Attends - Draft combine

- In the NBA players will attend a draft combine before getting picked out on a team. The team will decide if they want that player after seeing certain statistics of the player. This is connected by `Draft_combine_stats.csv` it has playerID to connect to the players.

- The Attends relationship is many-to-many since each player in the NBA will attend many draft combines over the seasons in NBA and each draft combine will have many attending players.

- In the attends relationship a player will partially participate in a draft combine since a player can be excused or not invited to a draft combine and a draft combine will totally participate since the draft combine will not be logged if the combine was cancelled.

- The primary key in this relationship is (Season and playerID) this is enough to uniquely identify returning players to a seasonal draft combine and players are distinguished from one and other

Player - Has - Player information

- A player in the NBA will have player information about their own biological metrics and other personal properties of themselves. Thus why this relationship exists. We can see that playerID is also in `common_player_info.csv`.

- The has relationship is one-to-one since every player will have only player information about themselves assigned to them and player information will be assigned to only the player that it coorsponds to and no one else. The data in `common_player_info.csv` relfects this playerID maps to one player information

- In the has relationship the player partially participates in the relationship since a player does not need to have their player information to exist in the database and the player information totally participates in the relationship since it is a weak entity and would not exist in the database without the actual player.

- The primary key in the relationship is only the playerID since the player information is a weak entity.