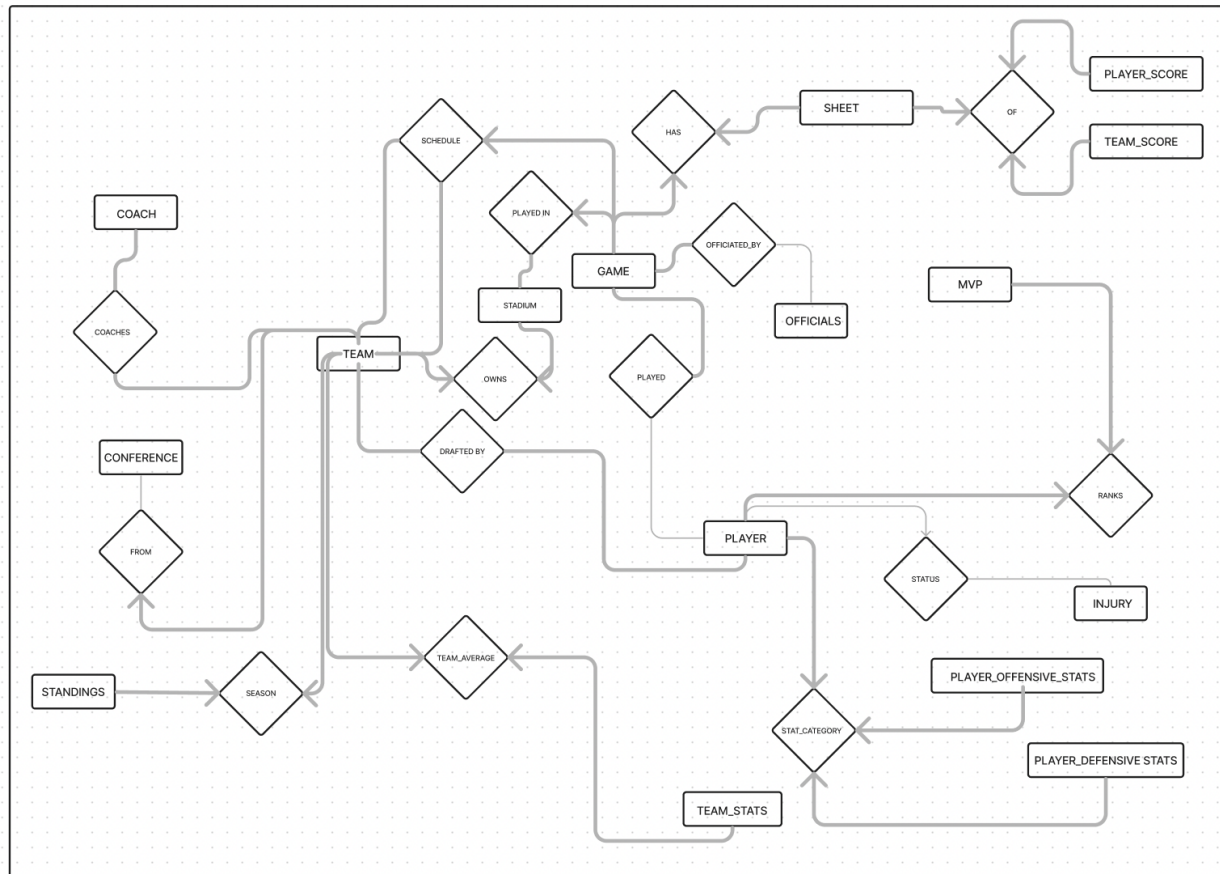


Data set link: [https://www.basketball-reference.com/leagues/NBA\\_2024\\_games-october.html](https://www.basketball-reference.com/leagues/NBA_2024_games-october.html)

The csvs are collected using a web-scraper.

ERD Diagram(The Diagram does not contain the attributes as it was getting cluttered, the attributes are mentioned below.)



### Description and Use case:

#### Entities:

Team, Player, Game, Conference, Coach, Standings, Stadium, Team\_Stats, Officials, Score\_Sheet, Player\_Score, Team\_Score, Player\_Ranking, Injury, Player\_stats

Relations(The ERD diagram represents the Participation and cardinality ratio):

Coaches (Team,Coach): Each Team is coached by a coach and a coach can have coached multiple teams

From (Team, Conference): Every Team belongs to a conference and every conference must have atleast one team

Season (Standings,Team): Each team has a single standing in a season, and each standing has a single team.

Team\_per\_game (Team\_stats,Team):Each Team has a single per game average stat and vice versa

Drafted\_by (Team,Player): A team can draft multiple players and a player can get drafted by multiple teams in a season.

Owns(Team,Stadium): Each team owns a single stadium and each stadium belongs to exactly one team.

Played\_in(Stadium,Game): Each game is played in a single stadium. A stadium will have multiple games played in a season.

Schedule(Team,Team,Game): Every team plays a game with the other teams as per the schedule.

Officiated\_by(Game,Officials): Each game has some officials, an official will officiate multiple games.

Boxscore(Game,Score\_sheet): Each game has a scoresheet and each scoresheet represents the score of that game.

Of(Score\_sheet, Player\_score\_sheet,Team\_score\_sheet): The scoresheet can either be of a team or a player.

Played(Player,Game): Many players play a game but not all players play a game. For example if the player is injured, they may not play a game.

Player\_per\_game (Player\_Stats): Every player has a per game average stat and vice versa

Status(Player,Injury): A player can be injured and the injury represent the status of a single player.

Ranks(Player,Player\_Ranking): Every player has a ranking and every ranking represents the rank of a single player.

### **Attributes:**

Team: Team\_ID (Primary Key), Team\_Name, Team\_Location

Player: Player\_ID (Primary Key), Player\_Name, Position

Game: Game\_ID (Primary Key),Date,Location

Conference: Conference\_ID (Primary Key),Conference\_Name

Stadium: Stadium\_ID (Primary Key),Stadium\_Name, Location

Standings: Standing\_ID (Primary Key),Team\_ID (Foreign Key),Wins,Losses

Team\_Stats:Stat\_ID (Primary Key),Team\_ID (Foreign Key)

Officials: Official\_ID (Primary Key),Official\_Name

Score\_Sheet: Score\_Sheet\_ID (Primary Key),Game\_ID (Foreign Key)

Player\_Score:Score\_ID (Primary Key),Score\_Sheet\_ID (Foreign Key),Player\_ID (Foreign Key),Points

Team\_Score: Score\_ID (Primary Key),Score\_Sheet\_ID (Foreign Key),Team\_ID (Foreign Key),Points

Player\_Ranking: Ranking\_ID (Primary Key),Player\_ID (Foreign Key),Rank

Injury: Injury\_ID (Primary Key), Player\_ID (Foreign Key),Description,Status

Player\_Stats: Stat\_ID (Primary Key),Player\_ID (Foreign Key)

Coaches:Team\_ID (Foreign Key),Player\_ID (Foreign Key),Coach\_Role

From: Team\_ID (Foreign Key),Conference\_ID (Foreign Key)

Season:Standing\_ID (Foreign Key),Team\_ID (Foreign Key)

Team\_per\_game: Stat\_ID (Foreign Key),Team\_ID (Foreign Key)  
Drafted\_by:Team\_ID (Foreign Key),Player\_ID (Foreign Key)  
Owns:Team\_ID (Foreign Key),Stadium\_ID (Foreign Key)  
Played\_in:Stadium\_ID (Foreign Key),Game\_ID (Foreign Key)  
Schedule:Team1\_ID (Foreign Key),Team2\_ID (Foreign Key),Game\_ID (Foreign Key)  
Officiated\_by:Game\_ID (Foreign Key),Official\_ID (Foreign Key)  
Boxscore:Game\_ID (Foreign Key),Score\_Sheet\_ID (Foreign Key)  
Of:Score\_Sheet\_ID (Foreign Key),Player\_score\_sheet\_ID (Foreign Key),Team\_score\_sheet\_ID (Foreign Key)  
Played:Player\_ID (Foreign Key),Game\_ID (Foreign Key),  
Player\_per\_game:Stat\_ID (Foreign Key),Player\_ID (Foreign Key)  
Status:Player\_ID (Foreign Key),Injury\_ID (Foreign Key)  
Ranks:Player\_ID (Foreign Key),Ranking\_ID (Foreign Key)

Write Up:

**Topic:** Blockchain technology and its integration into DBMS

Based on the recent ups and downs of the crypto industry, I decided to write about the development of Blockchain Technology. Blockchain can be a potential solution to the issues faced in the traditional DBMSs.

The current DBMSs are regulated by their respective organizations making them vulnerable to cyber-attacks and breaches. On the other hand, Blockchain provides a decentralized network where data is interlinked and acts a puzzle for which every piece is transparent and connected to each other hence ensuring an enhanced and secured database.

The blockchain technology could fill a huge gap in the DBMS by integrating cryptographical network and requiring authentication from the other nodes in the network. This could be a huge factor in the industries today where data integrity is crucial. Every transaction in the blockchain has a timestamp and a record which cannot be altered or fabricated, thus providing a transparent audit system and immune to risk and fraud. The Blockchain technology streamlines the process of any type of transaction by making it a peer-to-peer transaction and thus reducing redundancy and other transaction costs.

In today's Industry companies like. Grayscale, IBM, Financial Banks, and other such real world organizations have already started the transition to the Blockchain technology to integrate it into their existing DBMSs. Cyber Security companies have started working on the compatibility issues because of the traditional mainframes and storage systems and believe that one this issues are tackled, Databases will be more secure than ever.

Personally, I believe that the development of Blockchain could bring in a digital revolution in the field of databases and its security. Other than securing the database, the issues of Big Data and

their analysis could also go a long way and solve issues related to efficiency and data redundancy.