Project Proposal

By

Radhika Agarwal, Ayesha Mulla, Shreya Gajbhiye

Team Insight Squad

Sports Performance Analytics

The Scenario

The organization for which we are designing a database is the National Basketball Association (NBA), a professional basketball league renowned for hosting the world's most talented and famous basketball players and teams. This league operates a network of games, player development, and fan engagement across the United States and internationally.

The purpose of the NBA's database is to centralize and manage data related to league operations, including games (both league and practice matches), teams, players, coaches, and associated statistics. The database will serve multiple functions, such as recording and tracking the performance of players and teams, managing match schedules and locations, overseeing team and player affiliations, and facilitating fan engagement through viewership data. The database aims to support strategic decision-making and provide a rich source of information for analytics.

The intended users of the database are:

League Administrators: They use the database to manage the league's operations, including scheduling matches, overseeing team compositions, and enforcing league policies.

Coaches and Team Staff: They utilize player performance data to make strategic decisions about training and match tactics.

Players: They access their personal performance data for self-analysis and improvement.

Fans and Media: They derive insights from the statistics for entertainment and reporting purposes, although their access is more limited and typically through an API or front-end application.

Analysts and Statisticians: They delve into the data for performance analysis, trend prediction, and to provide insights to teams and the media.

Database Requirements:

The database comprises several core entities: Players, Teams, Coaches, Matches (League Matches and Practice Matches), Stadiums, and Stats. The relationship between these entities is as follows:

1. **Player**:
    - A player is identified by a unique player ID and has personal attributes like first name, last name, date of birth, height, weight, nationality, and the team they currently play for.
    - The attribute Age is a derived attribute that can be calculated using the date of birth attribute.
    - Each player belongs to exactly one team.
    - A player can be a mentor to another player, establishing a many-to-many recursive relationship indicating the mentorship between players.
    - A player has a detailed record of statistics for their performance in matches.
2. **Team**:
    - A team has a unique team ID, a name, and a home stadium.
    - A team can have a varying number of players, with a minimum of 5 and a maximum of 12.
    - Each team is coached by a coach.
    - Teams participate in league matches and can win league titles.

3. **Coach**:
    - Coaches has a unique coach ID, first name and the last name.
    - Coaches are associated with teams and have a start and potential end date for their contracts.
    - A coach can coach only one team at a time but may have coached different teams over time.

4. **Match**:
    - Each match has a unique match ID, a date, and a result.
    - A match entity is a superclass with two subclasses: a practice match and a league match.
    - **Practice match** has an intensity level which indicates the level of effort or competitiveness in the practice session.
    - **League match** has a viewership count a metric of the number of viewers that watched the match, which is vital for analyzing fan engagement and the popularity of games.

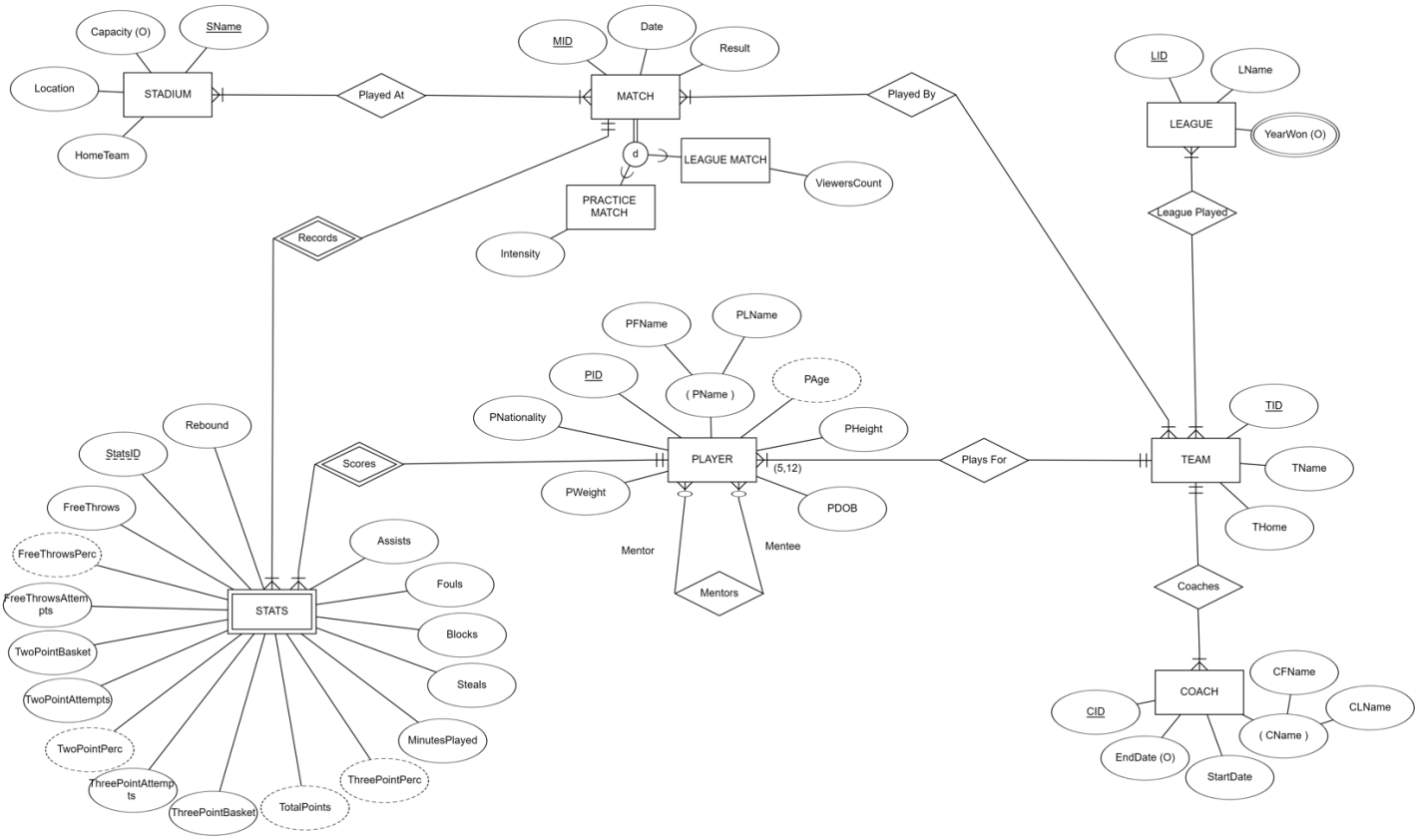
- A match is associated with a stadium where it is played. This is a many-to-one relationship, as many matches can be played in one stadium.
- A match involves two teams. This many-to-many relationship which links each match to the two teams that participated in it.
- A match has a detailed record of statistics for the performance of players in a particular match.

5. **Stadium**:
    - A stadium is identified by a name and the location.
    - A stadium has an optional attribute capacity that includes the seating capacity of each stadium.
    - Each stadium is associated with a home team and can host multiple matches.
6. **Stats**:
    - The STATS entity is a crucial component of the sports performance analytics system designed to record and analyze the statistical performance of players during matches.
    - Stats ID is a unique identifier for each set of statistics.
    - The stats entity records different quantitative attributes like Assists, Rebounds, Fouls, Blocks, Steal, MinutesPlayed, ThreePointBasket, ThreePointAttempts, TwoPointBasket, TwoPointAttempts, FreeThrows, FreeThrowsAttempts, etc. that helps in analyzing the performance of each player in a match.
    - We can also derive certain attributes such as FreeThrowPerc, ThreePointPerc, TwoPointPerc and TotalPoints from the above quantitative attributes.
    - Each set of stats is linked to a player and a match.

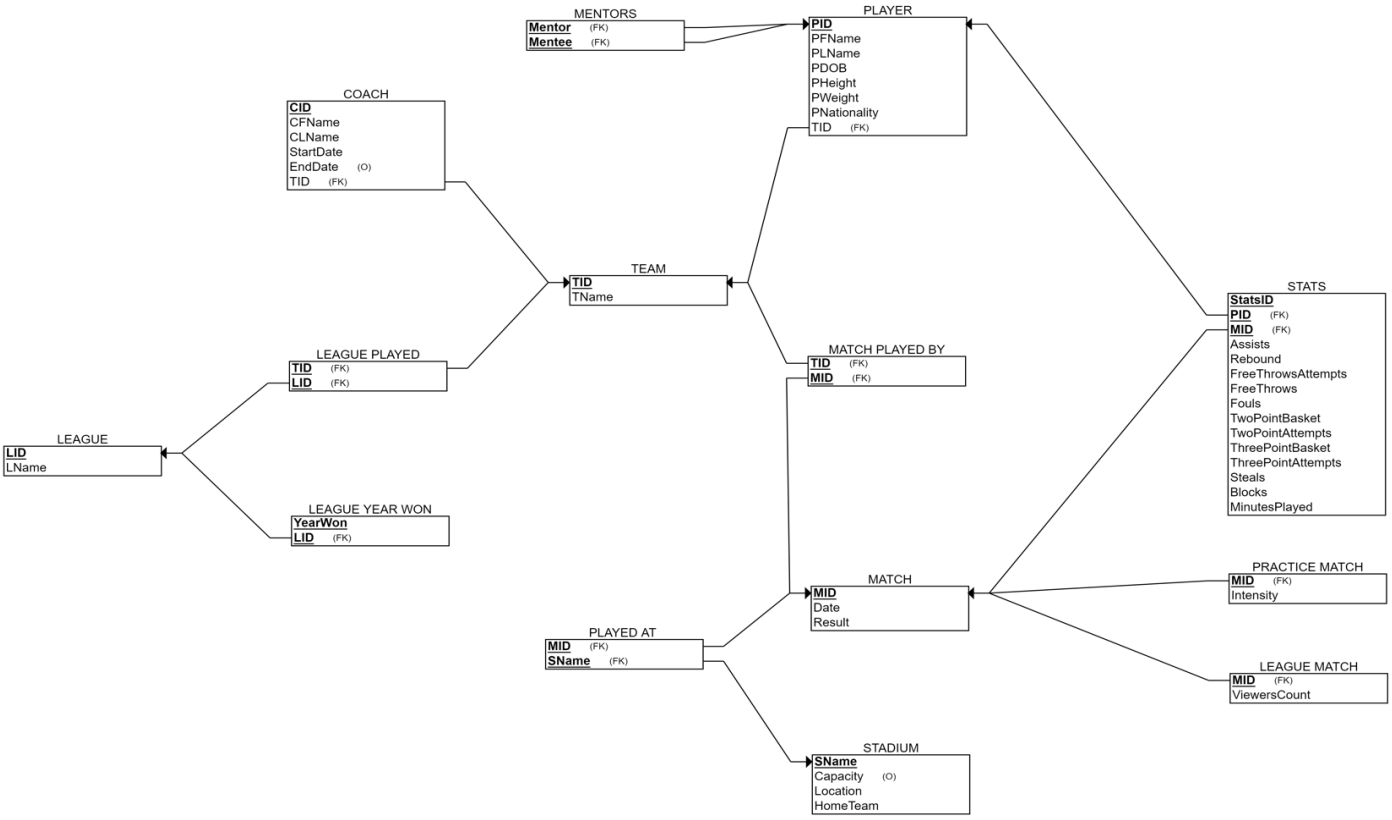
7. **League**:
    - The league entity keeps track of the different leagues within the NBA, including the year(s) each team has won the league.
    - Each League has a unique League ID and an official League Name.
    - A team can play multiple leagues and a League can be played by multiple teams.

Given these requirements, we can easily establish the necessary tables, relationships, and constraints to create a comprehensive database system that meets the NBA's needs. The database will be designed with scalability in mind, considering the league's potential expansion and the need to incorporate new types of data in the future.

Enhanced Entity Relationship Diagram



Relational Schema



Normalization Review:

All the tables in our relational schema are in 3NF form as there are no multivalued, functional dependencies or partial dependencies in the database.

Denormalization Review:

Denormalization involves reversing some of the normalization principles to improve database performance at the expense of some data redundancy and potential integrity issues.

Denormalizing the STATS Table:

Suppose we choose to denormalize the STATS table by combining it with player and match information to reduce the number of joins during queries.

The denormalized STATS table could include:

- StatsID
- PID

- MID
- TID
- Player Name (from PLAYER table)
- Match Date (from MATCH table)
- Team Name (from TEAM table)
- All statistical attributes (Assists, Rebounds, Points Scored, etc.)

Benefits of Denormalization:

- Query Simplification: Queries that need to report player statistics along with their names and match dates no longer need to join the STATS, PLAYER, and MATCH tables, simplifying query structure.
- Performance Improvement: Reducing the number of joins can significantly improve query performance, particularly for read-heavy databases where complex reporting is frequent.
- Ease of Use: For reporting tools or less complex application logic, having a single table that includes all related information can be easier to work with.

Drawbacks of Denormalization:

- Data Redundancy: Including player names and match dates in the STATS table means this information is duplicated across many rows, increasing storage requirements and the potential for inconsistency.
- Maintenance Overhead: Any updates to player names or match dates would need to be propagated to multiple rows in the STATS table which will increase the complexity of update operations.

Reverting the table back to 3NF:

To normalize the STATS table back to 3NF for the final project submission, we would remove the denormalized attributes (Player Name, Match Date, Team Name) and revert to using foreign keys (PID, MID) to reference the PLAYER and MATCH tables:

- StatsID (Primary Key)
- PID (Foreign Key)
- MID (Foreign Key)
- All statistical attributes (Assists, Rebounds, Points Scored, etc.)

By ensuring the table is back in 3NF, we eliminate redundancy, reduce the risk of data anomalies, and maintain the integrity of our database design. Thus, for performance requirements, we can normalize and denormalize tables in such a format.

Creation of the Database Schema:

We found all the data about teams and players from https://www.nba.com/

Following are some data sources to collect:

- Player information: https://www.kaggle.com/datasets/drgilermo/nba-players-stats
- Player and Team Statistics: https://www.kaggle.com/datasets/drgilermo/nba-players-stats
- Stadium Information: https://en.wikipedia.org/wiki/List_of_NBA_arenas
- All the player information along with the Match Statistics can be found here: https://www.basketball-reference.com/leagues/NBA_2016_per_game.html

Majority of data is collected from the above sources that help us structure our database. Following which we plan to customize our own data as per our requirements.

Data Types:

Choosing the right data types for database attributes is crucial for optimizing both storage and performance, and also to ensure that the data adheres to the intended constraints of its domain. Below are a few attributes with their specific datatypes that were chosen according to the schema.

- **CHAR:** TID, MID, CID, PID, LID are defined as fixed length identifiers which typically represent codes that do not change in length. Using CHAR for such fields is efficient when the length is always consistent, as it avoids the overhead associated with VARCHAR.
- **VARCHAR:** TName, CFName, CLName, LName, SName, PFName, PLName, PNationality, Intensity are variable in length as their actual content stored can differ significantly. VARCHAR is used here to save space because it only uses as much space as the content requires, plus an additional byte or two to store the length of the data.
- **INT**: Capacity, ViewersCount, Assists, Rebound, FreeThrowsAttempts, FreeThrows, Fouls, TwoPointBasket, TwoPointAttempts, ThreePointBasket, ThreePointAttempts, Steals, Blocks are numerical fields where the data represents whole numbers that is integers. Using INT for these attributes ensures that mathematical operations are straightforward and efficient.
- **FLOAT:** PHeight, PWeight, MinutesPlayed are assigned float as these fields might require decimal values to accurately represent measurements like height, weight, and time. FLOAT is suitable for these attributes because it can handle decimals and is adequate for the level of precision required here.
- **DATE:** Date, StartDate, EndDate, PDOB are fields that represent dates, requiring year, month, and day values. The DATE data type is ideal for storing points in time and is supported with date-specific functions in SQL that can be useful for extracting year or month and calculating differences between dates.

- **YEAR:** YearWon in LeagueYearWon relation is specifically storing years. Using the YEAR data type simplifies the storage and retrieval of year values, especially when only the year is relevant, and it uses less space than a full DATE type.

Overall, the selected datatypes help maintain data integrity, optimize database performance, and ensure that the storage requirements are suitable for the nature of the data being stored.

SQL QUERIES

Among the 10 queries executed, here are some queries with their detailed explanation.

1. Calculating the Stats per Player:
   This query calculates the average stats per player in all matches they've played, along with the total number of matches they've participated in.

QUERY:

```
SELECT
  p.PID,
  p.PFName,
  p.PLName,
  COUNT(mpb.MID) AS NumberOfMatches,
  AVG(s.Assists) AS AvgAssists,
  AVG(s.Rebound) AS AvgRebounds,
  AVG(s.FreeThrowsAttempts) AS AvgFreeThrowsAttempts,
  AVG(s.FreeThrows) AS AvgFreeThrows,
  AVG(s.Fouls) AS AvgFouls,
  AVG(s.TwoPointBasket) AS AvgTwoPointBaskets,
  AVG(s.ThreePointBasket) AS AvgThreePointBaskets,
  AVG(s.Steals) AS AvgSteals,
  AVG(s.Blocks) AS AvgBlocks,
  AVG(s.MinutesPlayed) AS AvgMinutesPlayed
FROM
  PLAYER p
JOIN
  STATS s ON p.PID = s.PID
JOIN
  MATCH_PLAYED_BY mpb ON s.MID = mpb.MID
GROUP BY
  p.PID, p.PFName, p.PLName;
```

OUTPUT:

PID	PFName	PLName	NumberOfMatches	AvgAssists	AvgRebounds	AvgFreeThrowsAttem...	AvgFreeThrows	AvgFouls	AvgTwoPointBaskets	AvgThreePoin...	AvgSteals	AvgBlocks	AvgMinutesPlayed
P00001	Curly	Armstrong	4	5.5000	4.5000	10.5000	6.0000	1.5000	5.0000	2.5000	2.0000	1.5000	35
P00002	Cliff	Barker	8	5.7500	10.0000	9.0000	5.7500	1.7500	9.5000	6.5000	5.0000	4.0000	55
P00003	Leo	Barnhorst	8	6.2500	9.5000	7.5000	4.5000	2.0000	9.0000	5.2500	5.0000	4.5000	51.25
P00004	Ed	Bartels	6	3.6667	8.3333	9.0000	4.0000	2.0000	9.6667	6.0000	5.0000	4.0000	55
P00005	Ralph	Beard	8	5.2500	9.7500	8.5000	6.0000	1.7500	10.0000	6.2500	6.0000	5.2500	58.75
P00006	Gene	Berce	2	15.0000	10.0000	5.0000	2.0000	0.0000	10.0000	7.0000	6.0000	6.0000	60
P00007	Charlie	Black	6	5.6667	11.6667	8.3333	5.3333	2.3333	11.6667	8.6667	6.3333	5.6667	65
P00008	Nelson	Bobb	2	2.0000	14.0000	11.0000	10.0000	1.0000	12.0000	9.0000	8.0000	8.0000	70
P00009	Jake	Bornheimer	6	8.0000	7.6667	6.0000	3.3333	2.3333	11.3333	8.0000	7.0000	6.0000	63.333333333333336
P00010	Vince	Boryla	8	4.5000	8.5000	8.5000	3.7500	2.0000	10.2500	7.0000	6.0000	5.2500	56.25
P00024	Johnny	Kerr	2	8.0000	12.0000	11.0000	8.0000	3.0000	14.0000	12.0000	8.0000	7.0000	75

How is it closely related to the purpose and intended users of our database?

This query is designed for basketball team managers and coaches who need a detailed understanding of their players' performance. It helps identify strengths and weaknesses across different metrics, which is crucial for making informed decisions on training priorities and game strategies.

The primary audience for this report is the coaching staff, including the head coach, assistant coaches, and possibly the team manager or sports analysts within the basketball organization.

Decisions based on this report could include:

- Which players need additional training in certain areas (e.g., free throws, three-point shooting).
- Identifying players with exceptional performance metrics for rewards or for strategic positioning during crucial matches.
- Evaluating players for potential transfers or contract renewals based on their performance trends.
- Adjusting match strategies to leverage players' strengths and mitigate weaknesses.

2. Calculating the Viewer Counts for each League

   This query calculates the average stats per player in all matches they've played, along with the total number of matches they've participated in.

   QUERY:

```sql
SELECT
  l.LName,
  SUM(lm.ViewersCount) AS TotalViewers
FROM
  LEAGUE l
INNER JOIN
  LEAGUE_PLAYED lp ON l.LID = lp.LID
INNER JOIN
  MATCH_PLAYED_BY mpb ON lp.TID = mpb.TID
INNER JOIN
  LEAGUE_MATCH lm ON mpb.MID = lm.MID
GROUP BY
  l.LName
ORDER BY
  TotalViewers DESC;
```

OUTPUT:

LName	TotalViewers
Women's National Basketball Association (WNBA)	71150
NBA	70150
Premier Basketball League	48000
Maximum Basketball League	47600
Basketball Africa League (BAL)	43000
Continental Basketball League	37600
NBA G League	36000
Drew League	32600
Tobacco Road Basketball League	32550
International Basketball League	23500
NBA 2K League	15000

How is it closely related to the purpose and intended users of our database?

This query is particularly useful for league administrators and broadcasters who need to understand viewer engagement across different leagues. Knowing which leagues attract the most viewers can help in making informed decisions regarding marketing strategies, broadcast scheduling, and promotional activities.

The primary audience for this report includes:

- League Administrators: To gauge the popularity of different leagues and to strategize league expansions or modifications.
- Marketing Teams: To tailor marketing campaigns based on the popularity of leagues.
- Broadcasters: To decide which leagues to prioritize for broadcasting based on viewers.

Decisions that might be influenced by this query include:

- Broadcasting Rights Negotiations: Leagues with high viewer counts might attract more lucrative broadcasting deals.
- Marketing and Promotions: Directing more resources towards the leagues with higher viewership to maximize revenue from advertisements and sponsorships.
- Scheduling: Adjusting match schedules to times that might attract more viewers based on historical data.
- Strategic Growth: Identifying potential areas for the introduction of new teams or expansion based on viewer interest in certain regions.

3. List All Matches Played at a particular Stadium:

   This SQL query is structured to retrieve detailed information about matches played at the "TD Garden" stadium.

QUERY:

```
SELECT
  mn.Date,
  mn.MID,
  mn.Result,
  s.SName AS StadiumName
FROM
  MATCH_NAME mn
INNER JOIN
  PLAYED_AT pa ON mn.MID = pa.MID
INNER JOIN
  STADIUM s ON pa.SName = s.SName
WHERE
  s.SName = 'TD Garden';
```

OUTPUT:

Date	MID	Result	StadiumName
2023-10-30	M00005	Home Win	TD Garden
2023-12-26	M00022	Away Win	TD Garden
2023-10-28	M00027	Home Win	TD Garden

How It Is Closely Related to the Purpose and Intended Users of our Database:

This query is vital for event coordinators, stadium management, and sports analysts who need precise data on matches held at a particular stadium. It helps them assess the frequency of events, evaluate logistical needs, and review the outcomes of games played at this location.

The primary audience for this report includes:

- **Stadium Managers and Staff**: To manage scheduling, maintenance, and staffing needs based on the frequency and timing of matches.
- **Event Coordinators**: To plan future events, considering the historical usage and outcomes of the venue.
- **Sports Analysts and Media**: To analyze match outcomes and prepare reports or broadcasts focused on events held at "TD Garden".

Decisions that might be influenced by this report include:

- **Event Scheduling and Planning**: Decisions about when to schedule future matches or other events based on the availability of the stadium.

- **Maintenance and Upkeep**: Scheduling maintenance around match dates to ensure the venue is in optimal condition for events.
- **Marketing and Promotions**: Tailoring marketing efforts to highlight key matches or events held at this venue to boost ticket sales and viewership.
- **Analytical Reporting**: Providing insights and data for sports analysts to use in discussions or reports about matches played at "TD Garden".

In summary, this query provides critical data for operational and strategic decision-making related to the management and utilization of "TD Garden". It supports efficient stadium operations and enhances the experience for teams, fans, and other stakeholders involved in the events held at the venue.

4. Top 5 Matches with Most Fouls:
   This query uses an aggregate function with a GROUP BY clause to find matches with the highest total number of fouls. This query provides a list of the matches with the highest number of fouls, including which team committed those fouls, sorted by the number of fouls descending.

   QUERY:

```sql
SELECT
  mn.Date,
  mn.MID,
  t.TName AS TeamName,
  SUM(s.Fouls) AS TotalFouls
FROM
  MATCH_NAME mn
JOIN
  STATS s ON mn.MID = s.MID
JOIN
  MATCH_PLAYED_BY mpb ON mn.MID = mpb.MID
JOIN
  TEAM t ON mpb.TID = t.TID
GROUP BY
  mn.Date, mn.MID, t.TName
ORDER BY
  TotalFouls DESC
LIMIT 5;
```

   OUTPUT:

Date	MID	TeamName	TotalFouls
2023-12-11	M00017	Indiana Pacers	5
2023-10-28	M00027	Philadelphia 76ers	5
2023-12-11	M00017	New York Knicks	5
2023-10-28	M00027	Boston Celtics	5
2023-12-17	M00006	New York Knicks	4

How It Is Closely Related to the Purpose and Intended Users of our Database:

This query is designed for referees, league officials, coaches, and sports analysts who are interested in understanding patterns of aggressive play or possible issues within teams or specific games.

The primary audience for this report includes:

- **League and Tournament Officials**: To monitor and address teams or matches with high foul rates, potentially influencing rule enforcement or disciplinary actions.
- **Coaches**: To identify and correct aggressive behavior in teams, aiming to reduce fouls in future games.
- **Sports Analysts**: To provide commentary or write articles about the discipline levels within leagues or teams.

Decisions influenced by this report might include:

- **Review and Adjustment of Game Strategies**: Coaches might use this information to adjust their team's play style to decrease fouls in future matches.
- **Disciplinary Measures**: League officials may consider additional training, fines, or other disciplinary measures for teams consistently appearing in this report.
- **Referee Training**: Referee committees might use this data to understand where more focus is needed during games or to improve foul calling accuracy.

Overall, this query supports critical assessments and decision-making processes in sports management, aiming to enhance the quality and fairness of gameplay.

5. Total number of matches won and lost by each team

   This SQL query efficiently calculates the total number of matches won and lost by each team across all games documented in the database.

   QUERY:

```sql
SELECT
  t.TName,
  COUNT(CASE WHEN (mn.Result = 'Home Win' AND st.HomeTeam = t.TID)
  OR (mn.Result = 'Away Win' AND st.HomeTeam != t.TID) THEN 1 END) AS MatchesWon,
  COUNT(CASE WHEN (mn.Result = 'Home Win' AND st.HomeTeam != t.TID)
  OR (mn.Result = 'Away Win' AND st.HomeTeam = t.TID) THEN 1 END) AS MatchesLost
FROM
  MATCH_NAME mn
JOIN
  MATCH_PLAYED_BY mpb ON mn.MID = mpb.MID
JOIN
  TEAM t ON mpb.TID = t.TID
JOIN
  PLAYED_AT pa ON mn.MID = pa.MID
JOIN
  STADIUM st ON pa.SName = st.SName
GROUP BY
  t.TName;
```

OUTPUT:

TName	MatchesWon	MatchesLost
Atlanta Hawks	3	2
Boston Celtics	3	2
Brooklyn Nets	3	3
Charlotte Hornets	0	1
Chicago Bulls	2	4
Cleveland Cavaliers	3	2
Detroit Pistons	2	3
Indiana Pacers	4	1
Milwaukee Bucks	3	3
New York Knicks	3	2
Philadelphia 76ers	1	4
Toronto Raptors	3	3

How is it closely related to the purpose and intended users of our database?

This query is crucial for league organizers, team managers, coaches, and sports analysts who need to evaluate team performance across a season or series of matches. It provides a clear view of each team's ability to win both at home and on the road, which is essential for strategic planning and analysis.

The primary audience for this report includes:

- **Team Coaches and Managers**: To assess the effectiveness of team strategies and player performance in different environments.
- **League Officials**: To maintain records and possibly use the information for seeding and qualification in tournaments.
- **Sports Analysts and Commentators**: To provide insights during broadcasts or in sports articles about teams' strengths and weaknesses.

Decisions that might be influenced by this query include:

- **Strategic Adjustments:** Teams may alter their training or game strategies based on their performance trends in home versus away games.
- **Player Development and Transfers**: Coaches might decide on player lineup changes or identify areas for improvement, and managers might consider player trades or acquisitions to strengthen weak areas.
- **Promotional and Marketing Decisions:** Marketing teams can use win/loss records to promote upcoming games, especially highlighting strong matchups or revenge games.