Comp 3380 Project NBA 2024-2025 Database

By: Dylan Beyak Userid: 7974864, Johnny Lee Userid:, Kameron Toews Userid:

Overwiew of project and summary of data

Introduction

This project is about a NBA Relational Database mainly focused on the season 2024-2025. This project includes Players, Teams, Games, Draft History, Coaches of teams and their stats for regular and playoff games, Arenas that NBA teams play in and Draft Combine history. It also shows how these main components of this database share relationships with one an other.

Summary Of Data

The reason we chose the Dataset was it had the correct relations of how we wanted to model the basketball database for what we wanted and an extensive amount of attributes to chose from and how we wanted to model our database. The total amount of rows that this dataset has is 37,262 rows. The list of attributes consisted of:

- Players personal information like height weight, position, Birthdate.
- Statsistics for players for each game they played like FG, 3P, BLK, etc.
- Coaches how many seasons they played and their career stats like wins,losses and total games for franchise current and overall games either in regular and in playoff.
- NBA Teams had their ID, team name team abbreviation, year founded.
- Games played in the 2024-2025 season had the teams that played against each other, the date, the arena that it was played in and the final scores.
- Draft history contained the drafts and first picks and overall picks.
- Draft combine history shows measurements of players and stats like bench press, body fat %, etc.

Loads of cleaning was required to get rid of excess columns that were not gonna be used in the database as well there were columns were not formatted in the right way an example was heights were showing up as years so there was a lot of cleaning that needed to be done. Here are all the sources that we had for our tables are the links below:

- Table for `Arena.csv` : https://geojango.com/pages/list-of-nba-teams?srsltid=AfmBOooHCjFL0n6ZRB-rIDjEjJ8x_ZAeYeU2I4F2A7WTUGfL7jPp9f40
- Coaches stats tables like `PlayoffGameCoachStats.csv`,`RegularGameCoachStats.csv` and `Coach.csv` came from : https://www.basketball-reference.com/leagues/NBA_2025_coaches.html
- Link to `PlayerInformation.csv`, `Drafts.csv`, `DraftCombine.csv`, `Player.csv` , `team.csv`, `Organization.csv`, `PlayerInformation.csv`: https://www.kaggle.com/datasets/eduardopalmieri/nba-player-stats-season-2425?resource=download
- For schedule table that was broken up into tables like `Games.csv`: https://www.basketball-reference.com/leagues/NBA_2025_games.html
- There is an additional PDF for the EER diagram since it is too large and you need to zoom.

Discussion Of Data Model

- The reason why it was broken down into these tables from the few tables we had was because it made the modeling and the database itself more clear and concise. If we had attributes that made sense to group together and didnt rely on the other attributes we wanted to split those tables up. There is logic as well behind the thought process like teams should be seperated from players and games but still share a relationship. Other examples like coaches and their stats were split up since we wanted basic information to be listed about coaches but not all their stats to follow along everytime we wanted access to just the basic information (this also applies for players and player information table). The rest of the other tables we found from the dataset were already sectioned off for us to use and clean.

- The only tricky decision around this dataset for modeling that was a huge issue is we only had data for games for season 2024-2025 so we had to work with our model only being one season

- Our model cleanly fit into the relational database since we made sure to list out all primary keys, foreign keys and all attributes in the ER model before converting it into the relational database.

- No, we do not regret the changes that we made in our model but there were decisions that we had to adjust. This was more at the later stage when it came to the queries. The first one being drafts is not enough to find what team is a player on most recently since trades happen in the NBA so we had to implement this by changing the data model and correctly implementing the change in the data. Secondly, the way we stored teamIDs in the games table we did not need to have a relationship between team and game since we can path our way through joining players.

- Yes there are a few places that the model could have been modeled differently. A couple examples of this include the following:

  - Combining the `Player` entity and the `PlayerInformation` entity, that did not need to be seperate.
  - The `Coach` entity and all the coaches stats like `PlayoffGameCoachStats` and `RegularGameCoachStats` these three tables could have been just one table and be filtered in queries.

Appendix

Dylan’s contributions

- Stage 1: Helped look for the datasets that connect to each other, Created the first timeline with deadlines, reviewed Johnny’s work for stage 1 and made any nessacery corrections to it

- Stage 2: Created Rough draft EER model in drawio and wrote the 1 paragraph reminder od chosen data, wrote all justifications rough draft

- Stage 3: Created the final EER model with Johnny and Kamerons feedback, finalized justfications of EER Model and did some of the final relational model, completed Merging,normalizing and cleaning in this stage as well with documentation of my steps.

- Stage 4: Wrote a few english queries while participating in review for Kameron’s stage providing any feedback if possible. Made sure the EER diagram was updated for stage 4

- Stage 5: Made sure the EER diagram was updated for any feedback given, did a few implementations of queries written in english and in SQL and wrote why an analyst might care about them. Participated in review again making sure final copy was well done.

- Stage 6: Helped think of the design of what the interface may look like. Wrote the SQL injection prevention plan. Participated in review making sure interface design was well structured and making any comments or adjustments as needed. Kept the EER model updated

- Final Project Report: Given queries for the interface to implement in Python. Started my account for authentication for where the database lives. Participated in review making sure any nessacery tweaks or bugs were fixed before final submission