CS 340 Project Step 2: Draft

Project Title: LEBRON
Team: Group 79 ("Table Talkers")
Team Members: Quinn Glenn and Tieg Zaharia

Fixes based on Feedback (see bottom of document) from Step 1:

- Removed junction tables from ERD
- Moved the M:N relationships to the tables that the junction tables point to, e.g. (from **PlayerPositions** to **Players** and **Positions**)
- Made **Season** and **SeasonPlayer** tables plural
- Added PKs to **PlayerPostions** and **SeasonPlayers** tables
- Added **ON DELETE** to some FKs to ensure we don't have orphaned rows / data
- Added missing auto_increment to **SeasonPlayers**
- Renamed **SeasonPlayers** to **SeasonTeamPlayers** to be more descriptive
- Normalization steps: no action needed.
    - NF1: all key attributes are defined, and there are no repeating groups
    - NF2: we have separated tables to ensure no partial dependencies, e.g. **SeasonTeamPlayers**
    - NF3: we have separated tables to avoid transitive dependencies, e.g. **Positions**

Project Outline

The NBA basketball league has over 500 NBA players across 30 different teams, and it claims tens of millions of fans in the USA alone. Sometimes those fans want to play a quick, casual game where their knowledge of the league is tested.

This website will solve that problem by generating quiz games for visitors in which they must guess an NBA player's name. The visitor is given hints based on various attributes like the conference they play in, their age, their team, and jersey number. The currently proposed tables contain data about the league itself (i.e. the game content), and the website will use full CRUD functionality on top of the database.

Database Outline

Players

- Purpose: stores detailed information about NBA players.
- Attributes:
    - playerID: int, auto_increment, unique, not NULL, PK
    - name: varchar(100), not NULL
    - age: int, not NULL
    - jerseyNumber: int, not NULL
- Relationship: each **Player** may have many **Positions** and vice versa, so a **M:N** relationship between Players and Positions is implemented on this table via the **playerID** and **positionID** FK columns
- Relationship: each **Player** may have played many Seasons and vice versa, so a **M:N** relationship between **Players** and **Seasons** is implemented on this table via the **playerID** and **seasonID** FK columns

Teams

- Purpose: records details about NBA teams.
- Attributes:
    - teamID: int, auto_increment, unique, not NULL, PK
    - name: varchar(100), not NULL
    - conferenceID: int, FK (references Conferences.conferenceID) ON DELETE SET NULL
- Relationship: each **Team** belongs to one **Conference**, so a **1:M** relationship is implemented via the **conferenceID** FK column

Conferences

- Purpose: stores information about NBA conferences.
- Attributes:
    - conferenceID: int, auto_increment, unique, not NULL, PK
    - name: varchar(50), not NULL

Positions

- Purpose: records details about the different positions in basketball.
- Attributes:
    - positionID: int, auto_increment, unique, not NULL, PK
    - name: varchar(50), not NULL
- Relationship: each **Player** may have many **Positions** and vice versa, so a **M:N** relationship between Players and Positions is implemented on this table via the **playerID** and **positionID** FK columns

PlayerPositions (M:N relationship table)

- Purpose: captures the many-to-many relationship between Players and Positions.
- Attributes:

    - playerPositionID: int auto_increment, unique, not NULL, PK
    - playerID: int, not NULL, FK (references Players.playerID) ON DELETE CASCADE
    - positionID: int, not NULL, FK (references Positions.positionID) ON DELETE CASCADE

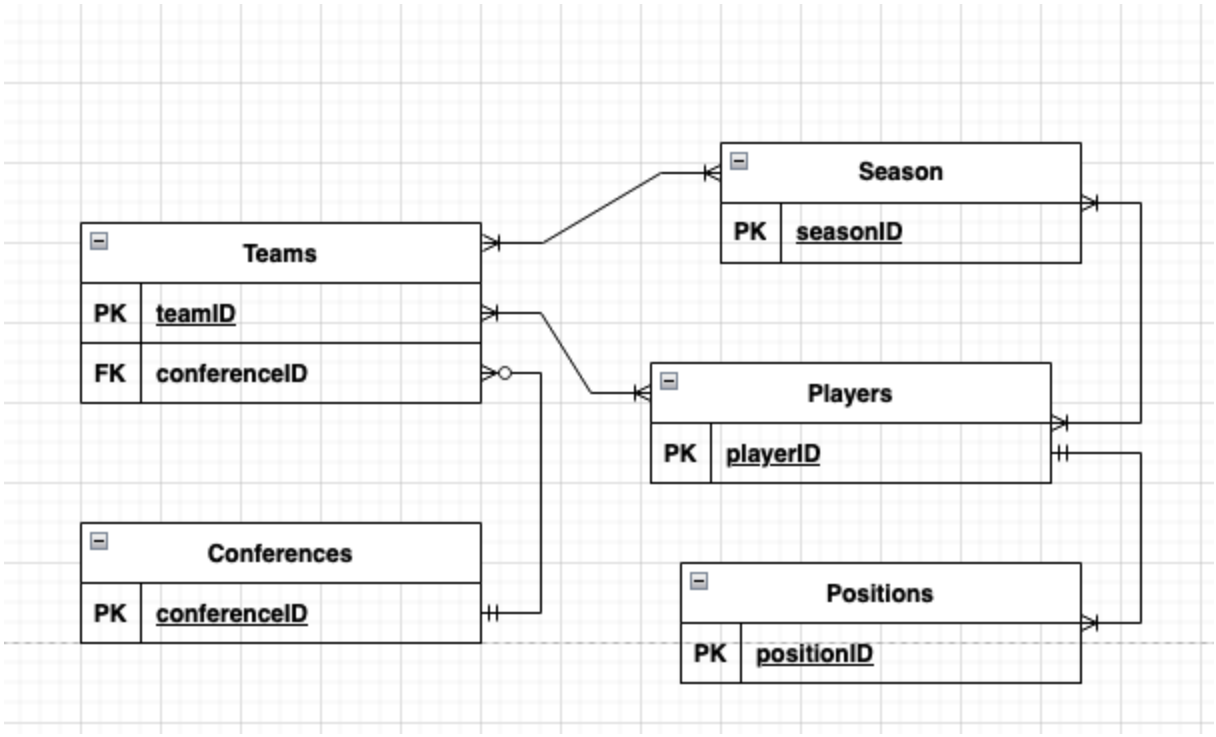
Seasons

- Purpose: stores information about an NBA season.
- Attributes:
    - seasonID: int, auto_increment, unique, not NULL, PK
    - year: int, not NULL, unique
- Relationship: each **Player** may have played many Seasons and vice versa, so a **M:N** relationship between **Players** and **Seasons** is implemented on this table via the **playerID** and **seasonID** FK columns

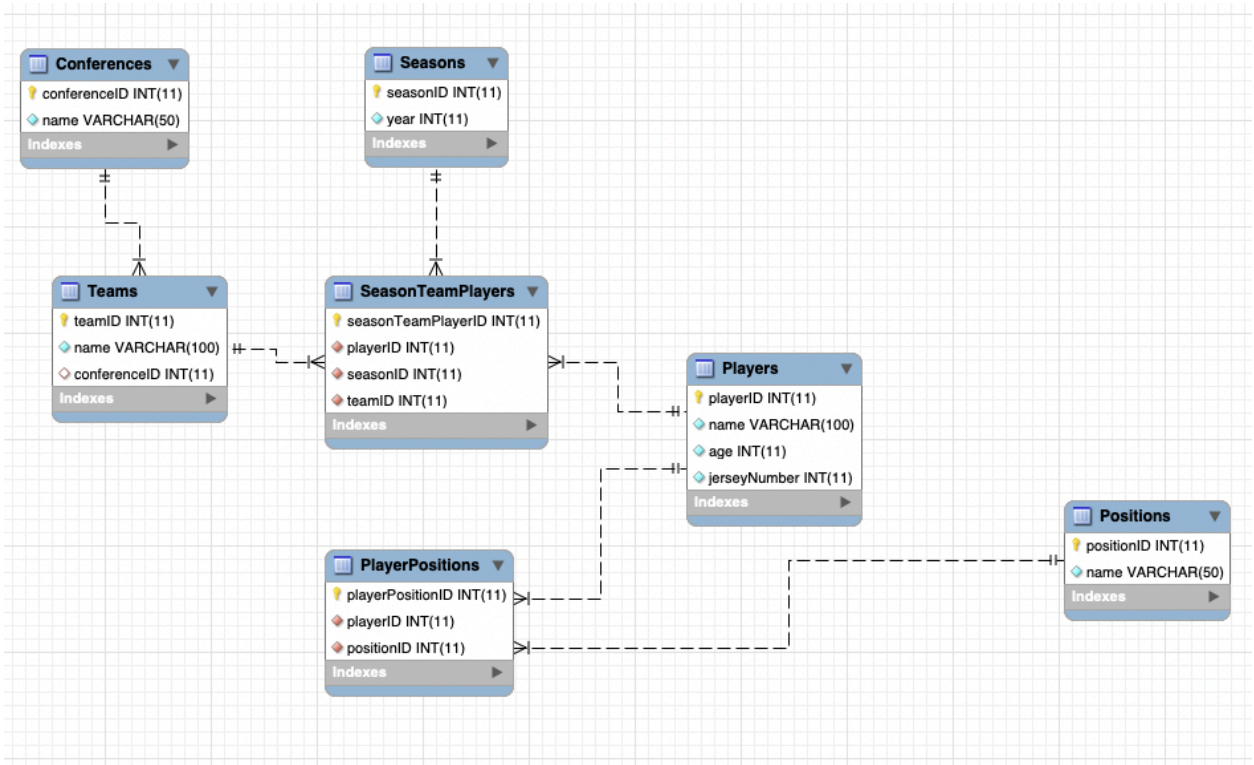
SeasonPlayers (M:N relationship table)

- Purpose: connects a Player to the Seasons they played
- Attributes:
    - seasonPlayerID: int, auto_increment not NULL, PK
    - playerID: int, not NULL, FK (references Players.playerID) ON DELETE CASCADE
    - seasonID: int, not NULL, FK (references Season.seasonID) ON DELETE SET NULL
    - teamID: int, not NULL, FK (references Team.teamID) ON DELETE SET NULL
- Relationship: each **Player** will belong to a single **Team** for a **Season**, so a **1:M** relationship between **SeasonPlayer** and **Team** is recorded via the **teamID** FK column

Entity-Relationship Diagram


Season
PK
seasonID
Teams
PK
teamID
FK
conferenceID
Players
PK
playerID
Conferences
PK
conferenceID
Positions
PK
positionID

Schema


Conferences
conferenceID INT(11)
name VARCHAR(50)
Indexes
Seasons
seasonID INT(11)
year INT(11)
Indexes
Teams
teamID INT(11)
name VARCHAR(100)
conferenceID INT(11)
Indexes
SeasonTeamPlayers
seasonTeamPlayerID INT(11)
playerID INT(11)
seasonID INT(11)
teamID INT(11)
Indexes
Players
playerID INT(11)
name VARCHAR(100)
age INT(11)
jerseyNumber INT(11)
Indexes
Positions
positionID INT(11)
name VARCHAR(50)
Indexes
PlayerPositions
playerPositionID INT(11)
playerID INT(11)
positionID INT(11)
Indexes

Example Data

Players

playerID	name	age	jerseyNumber
1	LeBron James	37	6
2	Stephen Curry	34	30
3	Kevin Durant	29	34
4	Luka Dončić	24	77

Teams

teamID	name	conferenceID
1	Los Angeles Lakers	1
2	Golden State Warriors	1
3	Brooklyn Nets	2
4	Milwaukee Bucks	2
5	Dallas Mavericks	1

Conferences

conferenceID	name
1	Western Conference
2	Eastern Conference

Positions

positionID	name
1	Point Guard
2	Shooting Guard
3	Small Forward
4	Power Forward
5	Center

PlayerPositions

playerPositionID	playerID	positionID
1	1	3
2	2	1
3	3	3
4	4	5
5	5	1

Seasons

seasonID	year
1	2023
2	2022
3	2021
4	2020
5	2019

SeasonTeamPlayers

seasonTeamPlayerID	playerID	seasonID	teamID
1	1	1	1
2	2	1	2
3	3	1	3
4	4	1	4
5	5	1	5

Step 1 Feedback

TA feedback:

“General Comments

1. ERD's generally do not include intersection/junction tables, instead are included within the schema diagram (learned in a future module)

Justin Dickerson , Jul 8 at 4:14pm

Corrective Comment

1. Regarding the SeasonPlayer intersection table within the ERD, check to make sure a PK is included.

Justin Dickerson , Jul 8 at 4:19pm”

1. SeasonPlayers is missing the auto_increment in the outline

2. Many of the entities only have 1 attribute. Consider combining them or adding additional attributes to the exiting ones to make each entity more distinct

3. Seasons to Players and Positions to Players should not be sharing the same relationship line, it makes the ERD less clear to understand

Devin Davies

Peer review feedback:

"*The project overview clearly addresses the problem of providing NBA fans with a fun and engaging way to test their knowledge through quiz games. The proposed website will generate quizzes where users guess NBA players' names based on hints like conference, age, team, and jersey number, demonstrating a need for a database backend. Specific facts about the number of NBA players (over 500) and teams (30) provide context for the project's scope.*

The draft outlines seven entities: Players, Teams, Conferences, Positions, PlayerPositions, Season, and SeasonPlayer. Each entity represents a distinct concept, ensuring a logical and organized database structure. For instance, Players capture detailed information about NBA players, while Teams record details about NBA teams. The purpose of each entity is well-articulated, with attributes listed alongside their data types and constraints.

The relationships between entities are well-defined. The 1:1 relationships, such as the one between Teams and Conferences where each team belongs to one conference, are correctly formulated. The draft includes two M:M relationships: one between Players and Positions through the PlayerPositions table, and another between Players and Seasons via the SeasonPlayer table, capturing the necessary data complexities.

The entity-relationship details present a logical and coherent view of the database. Naming conventions are consistent, with entities in plural form and attributes in singular form, following a clear pattern that aids readability.

Overall, the draft is comprehensive, with a clear problem description, detailed entity outlines, and correctly formulated relationships. The inclusion of both 1:1 and M:M relationships ensures a robust and logically structured database design. Explicitly mentioning all relationships, including 1:M, in the entity descriptions would add clarity. This project proposal outlines a promising approach to creating an engaging quiz game for NBA fans, supported by a well-designed database backend." - Dhaya Raja

"*Hello! It looks like your overview does accurately describe what problem is to be solved, and it has a lot of specific facts about the NBA and will be good to be solved by a database backend. You do have at least four entities, and each one has a coherent idea of what it's representing. Each of your outlines seems to accurately describe the constraints, data types, and relationships, along with the purpose. I didn't see a M:M in your description, but I think I spotted a couple in your diagram, your 1:M relationships seem to be properly outlined, but I'd make sure that you explicitly mention them in your description. Your ERD seems to be good, and there does seem to be consistency in your naming, entities, and capitalization as necessary. Overall, looks pretty good, I'd just make sure to explicitly define your M:M relationship somewhere! Seems like a pretty fun project idea, hope you all enjoy it!*" - Nathan Bernstein

"***Does the overview describe what problem is to be solved by a website with DB back end?***

    - *Yes, the website with DB backend will be used to generate quizzes for NBA fans regarding NBA player trivia. The quiz will be able to give fans hints based off of entity attributes for the generated questions.*
- ***Does the overview list specific facts?***
    - *There exist facts in the overview regarding the scale of the database application, and there are specific factual player attributes mentioned that will be used to give hints to fans taking the quiz.*
- ***Are at least four entities described, and does each one represent a single idea to be stored as a list?***
    - *The overview describes the application of five entities: players, teams, conferences, positions, and seasons. Each of these entities represents a distinct idea that can be stored as a list.*
- ***Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints, and describe relationships between entities?***
    - *The outline of entity details describes the purpose of each and lists attribute datatypes, but the presence of described entity relationships is inconsistent. The Players, Conferences, Positions, and Seasons entities do not have relationships indicated. Only the M:N relationship tables have relationships listed in the entity details outline.*
- ***Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?***

    - *There are two M:N relationships described and the ERD presents a logical view of the database. The 1:M relationships are also correctly formatted. Something to consider is that a quick internet search has yielded multiple team name changes and league realignments in the last two decades. I'm not sure what the temporal scope of your project is, but it may be wise to include some way to handle this and the 2004 league realignment wherein the conferences were adjusted.*
- ***Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?***
    - *a) There is consistency in naming between overview and entities/attributes.*
    - *b) Season and SeasonPlayer are two entities that should be made plural for consistency. All of the attributes are singular.*
    - *c) There is a consistent use of capitalization for naming attributes and entities (camel-case).*

“ - Michael Jagielski

“***Does the overview describe what problem is to be solved by a website with DB back end?***

Yes, the overview describes what problem is to be solved by a website with database backend, specifically that it holds data that will be utilized in generating questions for and evaluating responses to a quiz game about players in the NBA.

Does the overview list specific facts?

Yes, the overview lists specific facts relevant to design of the database. Specifically, it discusses that there are 500 NBA players across 30 teams, and data about these players is the subject-matter of the quiz game serviced by the database that will potentially be played by 10's of millions of NBA fans. The scale of the data and potential usage creates appropriate context for understanding (i) why a website with database backend is required and (ii) the parameters for design of the database.

Are at least four entities described, and does each one represent a single idea to be stored as a list?

Yes, the database outline describes a total of 7 entities, each of which represents a single idea to be stored as a list. Based on the overview description of the purpose of the database, the entities described and their associated attributes appear to capture all of the data necessary to support the proposed NBA player quiz. For example, one of the example scenarios provided in the overview states that quiz players will be provided hints as to the identity of the NBA player they are to guess / identify, which hints would include the NBA player's conference, age, team, and jersey number. Each of these data are included as attributes of a described entity, and the proposed relationships among entities appear sufficient to relate a specific player to each of these data.

Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints, and describe relationships between entities?

Yes, each entity described in the database outline includes a meaningful description of its purpose, identifies all attribute datatypes and constraints, and describes the relationships between entities. All entity attribute datatypes appear appropriate for the type of data associated with each attribute and all associated constraints seem appropriate, including several FK constraints related to 1:M or M:N relationships. In addition, as noted above, the scope and nature of the relationships described appear sufficient to relate a specific player to each of the data required to support the quiz game.

Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?

Yes, the 1:M relationships are correctly formulated. First, all locations where a 1:M relationship is identified are accurate. For instance, a 1:M relationship is identified between Teams and Conferences. This is appropriate, as a Team can only be part of one Conference, but a Conference has multiple teams. Second, the group correctly implemented the 1:M relationships by inclusion of the FK of each "one" entity (e.g., Conferences) as an attribute of each "many" entity (e.g., Teams).

With respect to M:N relationships, the database outline does include at least one M:N relationship, and all M:N relationships are correctly formulated. For example, the outline identifies a M:N relationship between Players and Positions, as each Player can have several

positions, and each Position is associated with multiple players. This relationship was correctly implemented via a separate intersection table containing FK's associated with both Players and Positions.

As a general matter, the ERD does present a logical view of the database, with (nearly) all tables and relationships accurately represented. One mistake I identified when reviewing the ERD is the relationship line between Players and PlayerPositions. The relationship correctly uses a 1:M relationship line; however, I believe the line is flipped. The "many" end of the line should be connected to PlayerPositions, and the "one" end should be connected to Players.

Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

Yes, the overview, entity descriptions, and ERD are all consistent. The group utilized a capitalized, plural naming convention for all entities and used a camelCase, singular naming convention for all entity attributes. The names as used in each section of the document are consistent.

In sum, the database overview, outline, and ERD are all well-formulated, with the single exception noted above with respect to the ERD. Based on the group's description of its intended use, I believe the proposed database will efficiently support an NBA player quiz game.

" - Cale Coffman