Hall of Fame Data Analysis

Group 4

Within this document we will provide a brief overview of the Hall of Fame database, its makeup and real world definition, and a high level overview of the real world application of the database. The Hall of Fame database aims to achieve the goal of providing an automated message delivery system that informs the proprietors of a basketball hall of fame when a player has become eligible for entry/acceptance into the hall of fame. Further, this database attempts to create and promote a standardization of basketball player metrics as well as having a defined standard for what constitutes as hall of fame worthy player statistics. As such this database will store not only player and team information, but additionally statistics for each player in order to facilitate its main function. The database would also be searchable for internal purposes to determine where a specific player is in relation to being accepted based off of their current statistics. Lastly, the database would be updated on terms determined by the hall of fame in order to keep data current up to whatever date they as clients specify as a requirement. This process can be automated or done manually depending on the preference of the client. To begin we will examine the conceptual model of the hall of fame database.

**ENTITIES**

1. Players

* Represents the individual players in the database
* Attributes: Name, Date of Birth, Position, height, age, weight, nationality.

1. Accolades

* Represents the accolades the individual player has accumulated throughout their career.
* Attributes: mvp status, rookie of the year, finals mvp, scoring title, defensive player of the year, championships, Olympic medals

1. Seasons

* Represents the seasons the player has participated in
* Attributes: year, description, made playoffs

1. Statis

* Represents the individual cumulative statistics for each player in the database.
* Attributes: points, assists, rebounds, steals, blocks, turnovers, minutes played, seasons played.

1. Teams

* Represents the team(s) that the individual played on as well as some information about the team.
* Attributes: team name, city, state

**Relationships**

1. Players-Seasons

* A player can play in more than one or many season
* A seasons statistics are unique to each player
* Type: One to Many

1. Players-Statistics

* A player can have no or many statistics.
* Statistics are unique to only one player.
* Type: One to Many

1. **Players-Teams**

* **A player can be on one or many teams**
* **A team can have one or many players**
* Type: Many to Many

1. Statistics-Accolodes

* A statistic can have one or many accolades
* An accolade can have one or many statistics
* Type: Many to Many

1. Accolades-Seasons

* An accolade can be a part of one or many seasons
* A season can have no or many accolades
* Type: Many to Many

Now moving from the conceptual model, we move over to the logical model to give a clearer view of how the tables and relationships in the database are established. This establishes what each attribute type is within each table in the database as well as providing a brief description of what the attribute signifies in terms of the data being stored within each one. We also include the establishment of primary and foreign keys within the database at this stage contending with the previously established relationships.

TABLES and their Attributes

For the makeup of the database we have a total of five tables to keep track of a variety of player metrics.

**Players**: Table to keep track of each players personal statistics such as name, position, or date of birth.

+player\_id: INT PRIMARY KEY

A unique id number generated for each player who is being stored in the data base

+team\_id: INT FOREIGN KEY

A unique id number generated for each team stored in the database. This is a foreign key to the Teams table.

+stat\_id: INT FOREIGN KEY

A unique id number generated for each statistic set that is bound to the player table through a foreign key.

+accolade\_id: INT FOREIGN KEY

A unique id number generated for each accolade set that is bound to the player through the use of a foreign key.

+ first\_name: VARCHAR (50)

The first name of the player stored as a varchar with a length of 50 characters.

+last\_name: VARCHAR(50)

The last name of the player stored as a varchar with a length of 50 characters.

+player\_dob: DATE

The date of birth of the player stored in the date format of DD/MM/YYYY.

+player\_position: VARCHAR(50)

The position(s) the player has played throughout the course of their career stored as a VARCHAR with a length of 50 characters.

+player\_height: FLOAT

The height of the player stored as a float with entry in feet and inches OR inches depending on client needs

+player\_weight: FLOAT

The weight of the player stored in pounds as a FLOAT.

+player\_nationality: VARCHAR(50)

The nationality of the player stored as a VARCHAR with a length of 50 characters.

**TEAMS**: Table to store the team information for the teams who are tracked by the database

+team\_id: INT PRIMARY KEY

An unique identifying number stored as an integer that identifies each team, is used as the primary key.

+team\_name: VARCHAR(50)

The name of the team stored as a VARCHAR with a length of 50 characters

+team\_city: VAR\_CHAR(50)

The city where the team is based stored as a VARCHAR with a length of 50 characters.

+team\_state: VAR\_CHAR(50)

The state in which the team is based stored as a VARCHAR with a length of 50 characters.

**SEASONS**: Table to store the season information for each team so that data can be organized by season or individual seasons can be examined.

+season\_id: INT PRIMARY KEY

The season has a unique identifier made up of a integer, seasons are unique to each player

+season\_year: INT

The season year stored as an integer to allow for four digit year input.

+season\_description: VARCHAR(255)

A brief description of the season for the player for taking note of exemplary accomplishments during the season, stored as a VARCHAR with a length of 255 characters.

+season\_made\_playoffs: BIT

A Boolean value to be marked as TRUE/FALSE for a given season to indicate if a player made it into the playoffs or not.

**ACCOLADES**: A table to store individual accolades for each player in order to keep track of all achievements made by the player throughout their career.

+accolade\_id: INT PRIMARY KEY

The accolade id is a unique integer generated for each player. Accolades are bound to the individual player through the player\_id.

+player\_id: INT FOREIGN KEY

The player id is a unique integer used to identify the individual player, functions as a foreign key in this table to bind to the player.

+accolade\_mvp\_status: BIT

A Boolean value to be marked as TRUE/FALSE to indicate whether or not a player has ever achieved MVP status

+accolade\_rookie\_of\_the\_year: SMALLINT

A raw count of the number of times the player has achieved the rookie of the year achievement.

+accolade\_finals\_mvp: SMALLINT

A raw count of the number of the times player has achieved the finals mvp achievement.

+accolade\_scoring\_title: SMALLINT

A raw count of the number of the times the player has achieved the scoring title achievement.

+accolade\_defensive\_poy: SMALLINT

A raw count of the number of times the player has achieved the defensive player of the year title.

+accolade\_champioships: SMALLINT

A raw count of the number of times the player has made it into a championship game

+accolade\_olympic\_medals: VARCHAR(255)

A list of the medals that the player has won at the Olympic level, listed as full words i.e (Gold, Silver, Bronze), stored as a VARCHAR with a length of 255 characters.

**STATIS**: The statistics table, used to store raw statistical data for each player like points scored or assist, which can be used with the seasons table to narrow the data to a specific date.

+stat\_id: INT PRIMARY KEY

A unique integer generated and assigned to each group of observations in the table for each player.

+player\_id: INT FOREIGN KEY

A unique integer generated for each player, used to bind the statistics for each player to the player table.

+season\_id: INT FOREIGN KEY

A unique integer generated for each season, used to be able to break the data down into each season.

+stat\_points: INT

The cumulative points the player has scored over their career stored as an integer.

+stat\_assist: INT

The cumulative number of assists the player has performed over their career stored as an integer.

+stat\_rebounds: INT

The cumulative number of rebounds the player has performed over their career stored as an integer.

+stat\_steals: INT

The cumulative number of steals the player has made over their career stored as an integer.

+stat\_blocks: INT

The cumulative number of blocks the player has made over their career stored as an integer.

+stat\_turnovers: INT

The cumulative number of turnovers the player has made over their career stored as an integer.

+stat\_minutes\_played: INT

The cumulative number of minutes the player spent playing on the court throughout their career stored as an integer.

+stat\_seasons\_played: INT

The cumulative number of seasons the player has participated in throughout their career stored as an integer.

Relationships (Foreign Keys)

1. Players

* player\_id: (INT, PRIMARY KEY, AUTO INCREMENT)
* team\_id: (INT, FOREIGN KEY REFERENCES Teams(team\_id)
* statis\_id: (INT, FOREIGN KEY REFERENCES Statis(stat\_id)
* accolade\_id: (INT, FOREIGN KEY REFERENCES Accolades(accolade\_id)

1. Accolades

* accolade\_id: (INT, PRIMARY KEY, AUTO INCREMENT)
* player\_id: (INT, FOREIGN KEY REFERENCES Players(player\_id)

1. Statis

* stat\_id: (INT, PRIMARY KEY, AUTO INCREMENT)
* player\_id: (INT, FOREIGN KEY REFERENCES Players(player\_id)
* season\_id: (INT, FOREIGN KEY REFERENCES Seasons(season\_id)

From here we will move to implement the models and documentation produced to create the database as described within this document. The further details of the project will be detailed in the Technical Document to accompany this Data Analysis.