

NBA Information Query and Management System Based on PHP and MySQL

Chang Zhou (zhou.chan@husky.neu.edu) and Song Jiang (jiang.son@husky.neu.edu)

Abstract:

NBA is widely considered to be the premier men's professional basketball league in the world. NBA fans need an easy way to obtain the up-to-date information of the league. Meanwhile, the administrators who manage and maintain the NBA database require an easy and efficient way to perform the task. Here, we designed and implemented a dynamic website with a MySQL database, which can be used to query and maintain NBA data. Our website is very easy to use and perfectly meets the needs of NBA fans and the administrators.

Introduction:

Recognized as the premier men's professional basketball league in the world[1], the National Basketball Association (NBA) has world-wide influence and hundreds of millions of fans. Currently there are 30 teams in the NBA league, which are divided into western and eastern conferences according to their geographical location. Each team has no more than 15 players. There are 82 regular matches during a typical regular season, and the 8 teams with best records in regular season in each conference will participate the play-offs. After three rounds of play-offs, the western champion and eastern champion will play against each other to win the NBA final championship.

The modern NBA matches have completed statistics, which includes the points, rebounds, assists, steals, blocks, turnovers, efficiency etc. NBA fans have a strong need to access to information of the league, for example, the result of a match, the performance of a team, and the stats of a favorite player. On the other hand, the administrator who manages all these information, requires an easy and efficient way to maintain the large amount of NBA data. Here, we designed and implemented an NBA information query and management system based on php and MySQL, which can support both the ordinary user (fans) to query the data, and the administrator to maintain the data. Our system contains a MySQL database which stores the information of the entire NBA league, including the teams, players, match schedules, scores, and statistics. Its specific form is a dynamic and easy-to-use website where users and administrators can query and/or insert, update, delete data by simply clicking a button.

Requirements: Our website has users and administrators. Users can query the data, while administrators can query, insert, update and delete data in the database. The detailed use cases are described as below.

Use cases by administrators:

1. Query the teams

Description:

An admin wants to query the information of teams, including basic information and team performance information.

Step-by-step description:

1. Admin: an administrator places a request to query the information of a specific team by clicking a link of that team
2. System: the system displays the information of that team on a web page

Explanation:

The team class has basic information like team name, location, championship and conference. The team class is linked with the team performance class. The team performance class has team performance information, like the scores per game and opponent's scores per game in the regular or play-off games.

2. Query the players

Description:

An admin wants to query the information of players, including basic information and performance statistics.

Step-by-step description:

1. Admin: an administrator places a request to query the information of players by entering the player's name
2. System: the system returns a list of players that match the name
3. Admin: the administrator selects a player by clicking the player's name
4. System: more information about that selected player is displayed on a new web page

Explanation:

The player class has basic information including player name, date of birth, the year when that player enters NBA, player height, player weight, player position. The player class is linked with the player performance class. The player performance class has player performance information, like points per game, rebounds per game, assists per game, steals per game and blocks per game in the regular or play-off games.

3. Delete a player

Description:

An admin wants to delete a player from a team if the player is retired or fired or traded to another team.

Step-by-step description:

1. Admin: an administrator deletes a player from a team
2. System: the system returns a confirmation message and deletes the player from the specified team.

Explanation:

The player class is linked with the team class and has a foreign key referencing to team id. Deleting a player from a team is to delete the foreign key.

4. Insert a player

Description:

An administrator wants to insert a player into a team if a player is traded to the team or is employed by the team.

<p>Step-by-step description:</p> <ol style="list-style-type: none"> 1. Admin: an administrator inserts a player to a team 2. System: the system returns a confirmation message and inserts a player to a specified team
<p>Explanation:</p> <p>The player class is linked with the team class and has a foreign key referencing to team id. Inserting a player into a team is to insert the foreign key to the team id.</p>

5. Update the match performance of a player

<p>Description:</p> <p>An administrator wants to update the match performance of a player if he attends a new match in the season.</p>
<p>Step-by-step description:</p> <ol style="list-style-type: none"> 1. Admin: an administrator updates the information of match performance of a player 2. System: the system returns a confirmation message and updates the match performance of the player
<p>Explanation:</p> <p>The player match performance class is linked with the player class and has a foreign key referencing to player id. Given a player id, the system can modify relative records in player performance table.</p>

6. Insert the match performance of a player in the all-star game

<p>Description:</p> <p>An administrator wants to insert the match performance of a player in the all-star game if the player is selected to attend the all-star game.</p>
<p>Step-by-step description:</p> <ol style="list-style-type: none"> 1. Admin: an administrator inserts the information of match performance of a player in the all-star game 2. System: the system returns a confirmation message and inserts the match performance of the player in the all-star game
<p>Explanation:</p> <p>The player match performance class is linked with the player class and has a foreign key referencing to player id. Given a player id, the system can insert relative records in player match performance table.</p>

7. Update the match performance of a team

<p>Description:</p> <p>An administrator wants to update the match performance of a team if it has a new match in the season.</p>
<p>Step-by-step description:</p> <ol style="list-style-type: none"> 1. Admin: an administrator updates the information of match performance of a team 2. System: the system returns a confirmation message and updates the match performance of the team

Explanation:

The team match performance class is linked with the team class and has a foreign key referencing to team id. Given a team id, the system can modify relative records in team match performance table.

8. Update the coach of a team

Description:

An administrator wants to update the coach of a team if the team replaces its coach in the season.

Step-by-step description:

1. Admin: an administrator updates the coach of a team
2. System: the system returns a confirmation message and updates the coach of the team

Explanation:

The coach information is in the Coach class. Coach class is linked to the Team class and has a foreign key referencing to team id. Updating the coach of a team is to update the foreign key.

9. Insert the honor of a player

Description:

An administrator wants to insert an record of an honor of a player if he obtains the honor in the season.

Step-by-step description:

1. Admin: an administrator inserts the honor of a player
2. System: the system returns a confirmation message and updates the honor of the player

Explanation:

The honor information is in the Honor class. Honor class is linked to the Player class and has the foreign key referencing to player id. Inserting the honor of a player is to insert the foreign key.

Use cases by users:

10. Query the players

Description:

A user wants to query the information of players, including basic information and match information.

Step-by-step description:

1. User: a user places a request to query the information of players
2. System: the system returns the information of players that matches the query

Explanation:

The player class has basic information including player name, date of birth, the year when that player enters NBA, player height, player weight, player position. The player class is linked with the player performance class. The player performance class

has player performance information, like points per game, rebounds per game, assists per game, steals per game and blocks per game in the regular or play-off games.
--

11. Query the teams

Description: A user wants to query the information of teams, including basic information and match information.
Step-by-step description: 1. User: a user places a request to query the information of teams 2. System: the system returns the information of teams that matches the query
Explanation: The team class has basic information like team name, location, championship and conference. The team class is linked with the team performance class. The team performance class has team performance information, like the scores per game and opponent's scores per game in the regular or play-off games.

12. Query the coaches

Description: A user wants to query the information of coaches.
Step-by-step description: 1. User: a user places a request to query the information of coaches 2. System: the system returns the information of coaches that matches the query
Explanation: The coach information is in the Coach class which is linked to the Team class. The Coach class has the name and date of birth of a coach. The Team class has the name of the team.

Design:

Front-end design:

The website contains a home page and several child pages that provide information about NBA teams, players, coaches, schedules&scores, standings, and personal honors. These child pages can be reached through the navigation bar on the home page. On the teams page, the names of all 30 NBA teams are listed, and users can click a name to open a new page which shows relative information of that team, including its location, the head coach of the team, the NBA championships won by the team, the seasonal performance, and a list of the players. On the players page or the coaches page, there is a text box where users can enter a name of a player or a coach and query that player or coach. The schedule&scores page shows the results of the most recent games and the schedules of unmatched games. The standings page lists the current rankings of the western and eastern conference. The personal honors pages contains information about the honors won by players.

While ordinary users can only query the data, administrators can maintain the data after log in. Here we did not implement the log in authentication function since it was beyond the objectives of this project. However, we provided a log in link which links to the administrator

page. On this page, the administrator can not only query all the information as an ordinary user do, but also insert, update or delete the information.

Database design:

The database design can be illustrated by the UML diagram as below:

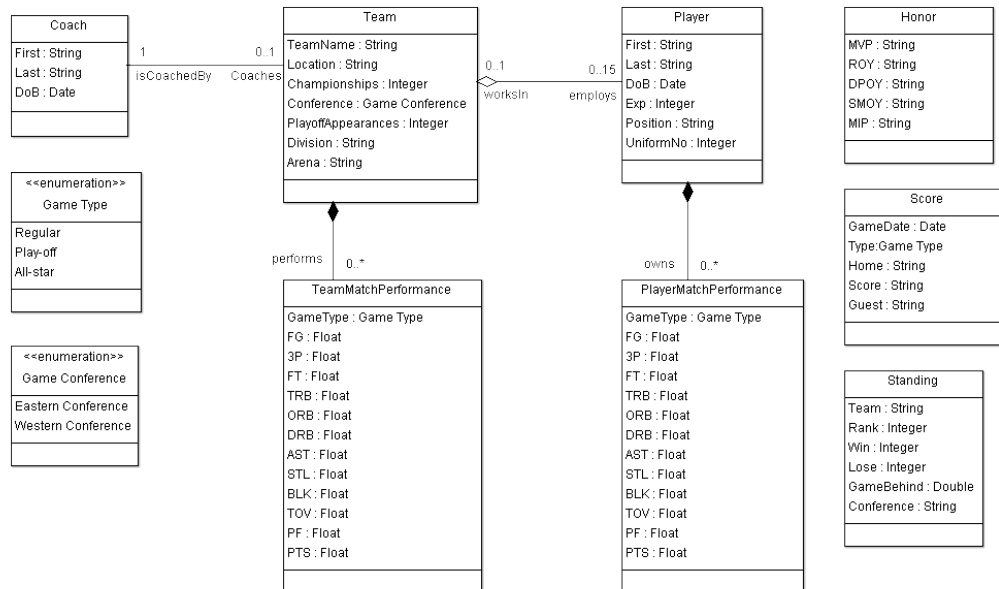


Figure 1. The UML diagram showing the relationship of the tables in the NBA database of this project.

The steps to create a database using MySQL on the platform of phpMyAdmin.

1. Create a database named NBA.

Databases

Create database

NBA Collation Create

2. Add tables and attributes in the database.

Create table

Name: Number of columns:

Name	Type	Length/Values
<input type="text" value="id"/>	<input type="text" value="INT"/>	<input type="text" value="20"/>
Pick from Central Columns		
<input type="text" value="Name"/>	<input type="text" value="VARCHAR"/>	<input type="text" value="255"/>

3. Download NBA data from [5] <http://www.basketball-reference.com>.

Current Roster

[Glossary](#) · [SHARE](#) · [Embed](#) · [CSV](#) · [Export](#) · [PRE](#) · [LINK](#) ·

No.	Player	Pos	Ht	Wt	Birth Date	Exp	College
0	Avery Bradley	SG	6-2	180	November 26, 1990	 5	University of Texas at Austin
7	Jared Sullinger	PF	6-9	260	March 4, 1992	 3	Ohio State University
44	Tyler Zeller	C	7-0	253	January 17, 1990	 3	University of North Carolina
41	Kelly Olynyk	C	7-0	238	April 19, 1991	 2	Gonzaga University
4	Isaiah Thomas	PG	5-9	185	February 7, 1989	 4	University of Washington
36	Marcus Smart	PG	6-4	220	March 6, 1994	 1	Oklahoma State University
13	James Young	SG	6-6	215	August 16, 1995	 1	University of Kentucky
11	Evan Turner	SG	6-7	220	October 27, 1988	 5	Ohio State University
12	Terry Rozier	PG	6-2	190	March 17, 1994	 R	University of Louisville
28	R.J. Hunter	SG	6-5	185	October 24, 1993	 R	Georgia State University
99	Jae Crowder	SF	6-6	235	July 6, 1990	 3	Marquette University
8	Jonas Jerebko	PF	6-10	231	March 2, 1987	 5	
90	Amir Johnson	PF	6-9	240	May 1, 1987	 10	
55	Jordan Mickey	PF	6-8	235	July 9, 1994	 R	Louisiana State University
30	John Holland	G-F	6-5	205	November 6, 1988	 R	Boston University

4. Process and rearrange the data in the same order with the order of attributes in the corresponding tables

#	Name	Type
<input type="checkbox"/> 1	id	int(40)
<input type="checkbox"/> 2	gameDate	date
<input type="checkbox"/> 3	type	varchar(225)
<input type="checkbox"/> 4	home	varchar(225)
<input type="checkbox"/> 5	score	varchar(255)
<input type="checkbox"/> 6	guest	varchar(255)

id	gameDate	type	home	score	guest
1	4/11/16	Regular	Houston Rockets	130:110	Los Angeles Lakers
2	4/11/16	Regular	Washington Wizards	113:98	Charlotte Hornets
3	4/11/16	Regular	Los Angeles Clippers	98:91	Dallas Mavericks
4	4/11/16	Regular	Denver Nuggets	84:100	Utah Jazz
5	4/11/16	Regular	Philadelphia 76ers	108:109	Milwaukee Bucks

5. Import the data into the database: Select one table → Import → Choose File → Format: CSV using LOAD DATA → Columns separated with: “,” → Use LOCAL keyword → Go

[Browse](#) [Structure](#) [SQL](#) [Search](#) [Insert](#) [Export](#) [Import](#)

File may be compressed (gzip, bzip2, zip) or uncompressed.

A compressed file's name must end in **[format].[compression]**. Example: **.sql.zip**

Browse your computer: no file selected

(Max: 128MiB)

You may also drag and drop a file on any page.

Character set of the file:

Partial import:

☒ Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. (*This break transactions.*)

Skip this number of queries (for SQL) or lines (for other formats), starting from the first one:

Other options:

☒ Enable foreign key checks

Format:

Format-specific options:

☐ Update data when duplicate keys found on import (add ON DUPLICATE KEY UPDATE)

Columns separated with:

Columns enclosed with:

Columns escaped with:

Lines terminated with:

Column names:

☐ Do not abort on INSERT error

☒ Use LOCAL keyword

Go

Implementation: The web pages are written in HTML, CSS and php. The layout of the web pages are based on an example from codacademy.com[2]. The php syntax can be found on w3schools.com[3]. The database is supported by MySQL. The development enviroment is XAMPP for Windows v5.6.19[4].

Discussion: All use cases described in the requirements have been fulfilled. Besides that, our system provides additional functionalities such as query information about schedules and scores. The point-to-point explanations are as below.

Use cases by administrators:

1. Query the teams

Description:

An admin wants to query the information of teams, including basic information and team performance information.

Demonstration:

1. Admin: click the “log in” link on the home page which links to the admin page
2. System: the names of all 30 NBA teams are listed on the admin page
3. Admin: select a team by clicking its name
4. System: information of that selected team are displayed in a new page

2. Query the players

Description:

An admin wants to query the information of players, including basic information and performance statistics.

Demonstration:

1. Admin: click the “log in” link on the home page which links to the admin page
2. System: a text box is displayed on the admin page where the administrator can enter the name (first name, last name or full name) of a player for the query

3. Admin: enter the name and click the “search” button
4. System: all players whose name matches the entered name are listed below the text box
5. Admin: select a player from the list by clicking the player’s name
6. System: more information about that player is displayed on a new web page

3. Delete a player

Description:

An admin wants to delete a player from a team if the player is retired or fired or traded to another team.

Demonstration:

1. Admin: log in, query and go to the player’s page as described above
2. System: the player’s basic information is displayed and a row of text boxes with an “update” button are shown under the information
3. Admin: in the text box which corresponding to the player’s team, enter the name of a new team, or “free player” if there is no new team, then click the “update” button
4. System: a confirmation message will pop up and the player’s team will be updated to the new team or “free player”

4. Insert a player

Description:

An administrator wants to insert a player into a team if a player is traded to the team or is employed by the team.

Demonstration:

1. Admin: log in, query and go to the player’s page as described above
2. System: the player’s basic information is displayed and a row of text boxes with an “update” button are shown under the information
3. Admin: in the text box which corresponding to the player’s team, enter the name of a new team, or “free player” if there is no new team, then click the “update” button
4. System: a confirmation message will pop up and the player’s team will be updated to the new team or “free player”

5. Update the match performance of a player

Description:

An administrator wants to update the match performance of a player if he attends a new match in the season.

Demonstration:

1. Admin: log in, query and go to the player’s page as described above
2. System: the player’s performance information is displayed and a row of text boxes with an “update” button are shown under the information
3. Admin: enter new data into the text boxes, and click the “update” button
4. System: a confirmation message will pop up and the player’s performance will be updated

6. Insert the match performance of a player in the all-star game

Description:

An administrator wants to insert the match performance of a player in the all-star game if the player is selected to attend the all-star game.

Demonstration:

1. Admin: log in, query and go to the player's page as described above
2. System: the player's performance information is displayed and a row of text boxes with an "update" button are shown under the information
3. Admin: in the box which corresponding to the game type, enter "all-star"; then enter new stat data into the rest text boxes, and click the "update" button
4. System: a confirmation message will pop up and the player's performance in a all-star game will be added into the database

7. Update the match performance of a team

Description:

An administrator wants to update the match performance of a team if it has a new match in the season.

Demonstration:

1. Admin: log in and go to the team's page as described above
2. System: the team's performance information is displayed and a row of text boxes with an "update" button are shown under the information
3. Admin: enter new data into the text boxes, and click the "update" button
4. System: a confirmation message will pop up and the team's performance will be updated

8. Update the coach of a team

Description:

An administrator wants to update the coach of a team if the team replaces its coach in the season.

Demonstration:

1. Admin: log in and go to the team's page as described above
2. System: the team coach information is displayed and a row of text boxes with an "update" button are shown under the information
3. Admin: enter new data into the text boxes, and click the "update" button
4. System: a confirmation message will pop up and the coach of the team will be updated

9. Insert the honor of a player

Description:

An administrator wants to insert an record of an honor of a player if he obtains the honor in the season.

Demonstration:

1. Admin: click the "log in" link on the home page which links to the admin page
2. System: a section named "manage player honors" is displayed

3. Admin: enter the player's full names into the text boxes and click the "update" button
4. System: a confirmation message pops up indicating either the honor record has been updated or the player names entered are not exist.

Use cases by users:

10. Query the players

Description:

A user wants to query the information of players, including basic information and match information.

Demonstration:

1. User: click the players link on the home page which links to the players page
2. System: a text box with a "search" button are shown on the players page
3. User: enter a name (first, last or full name) and click "search"
4. System: all players whose name matches the entered name are listed in a table
5. User: select a player by clicking the player's name
6. System: a new page will pop up which displays more information of that player

11. Query the teams

Description:

A user wants to query the information of teams, including basic information and match information.

Demonstration:

1. User: click the teams link on the home page which links to the teams page
2. System: the names of all 30 NBA teams are listed on the teams page
3. User: select a team by clicking its name
4. System: information of that selected team are displayed in a new page

12. Query the coaches

Description:

A user wants to query the information of coaches.

Demonstration:

1. User: click the players link on the home page which links to the players page
2. System: a text box with a "search" button are shown on the players page
3. User: enter a name (first, last or full name) and click "search"
4. System: all coaches whose name matches the entered name are listed in a table
5. User: select a coach by clicking the coach's name
6. System: a new page will pop up which displays more information of that coach

Additional use cases provided by our system:

13. Query the schedules&scores

Description:

A user wants to query the schedules and scores

Demonstration:

1. User: click the schedules&score link on the home page which links to the schedules&score page
2. System: the most recent schedules and scores are listed on the page

14. Query the standings

Description:

A user wants to query the conference standings

Demonstration:

1. User: click the standings link on the home page which links to the standings page
2. System: the current standings of each conferences are listed on the page

15. Update the schedules&scores

Description:

An Administrator wants to update the score of a game

Demonstration:

1. Admin: log in to the admin page as described above
2. System: a section named “manage scores” is displayed on the page
3. Admin: enter the date of a game and click “search” button
4. System: all games on that date are listed and for each game, there is a text box where the administrator can enter the score for that game
5. Admin: enter the score of a game and click “update” button
6. System: a confirmation message will pop up and the score of that game is updated.

Conclusion: In this project, we designed and implemented a dynamic website with a MySQL database, which can be used to query and maintain NBA data. The website fulfilled all use cases requirements. Future work could be carried out to introduce more functionalities to the website, such as providing tickets booking service. Efforts should also be put to improve the security of the website and the database.

References:

- [1] https://en.wikipedia.org/wiki/National_Basketball_Association
- [2] <https://www.codecademy.com/learn/make-a-website>
- [3] <http://www.w3schools.com/php/>
- [4] <https://www.apachefriends.org/index.html>
- [5] <http://www.basketball-reference.com>