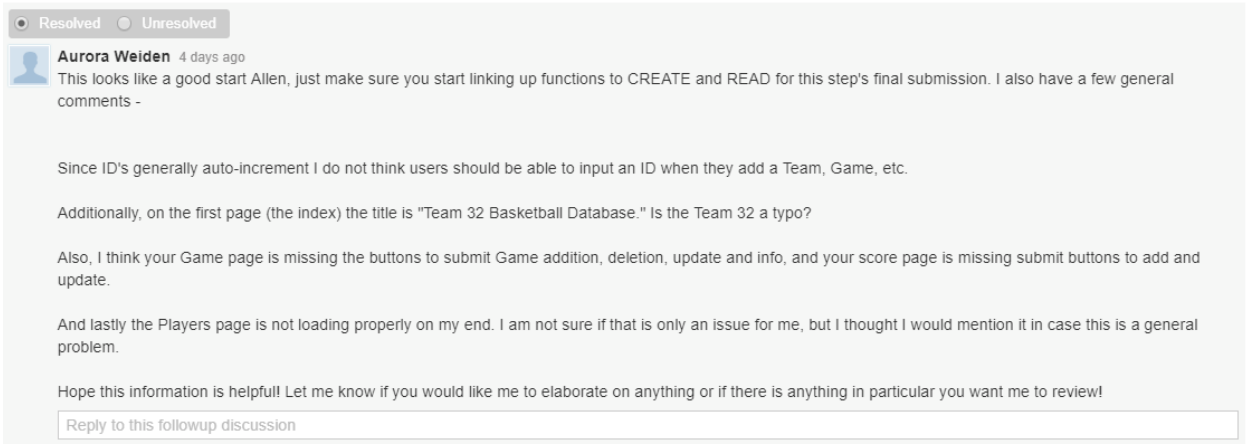
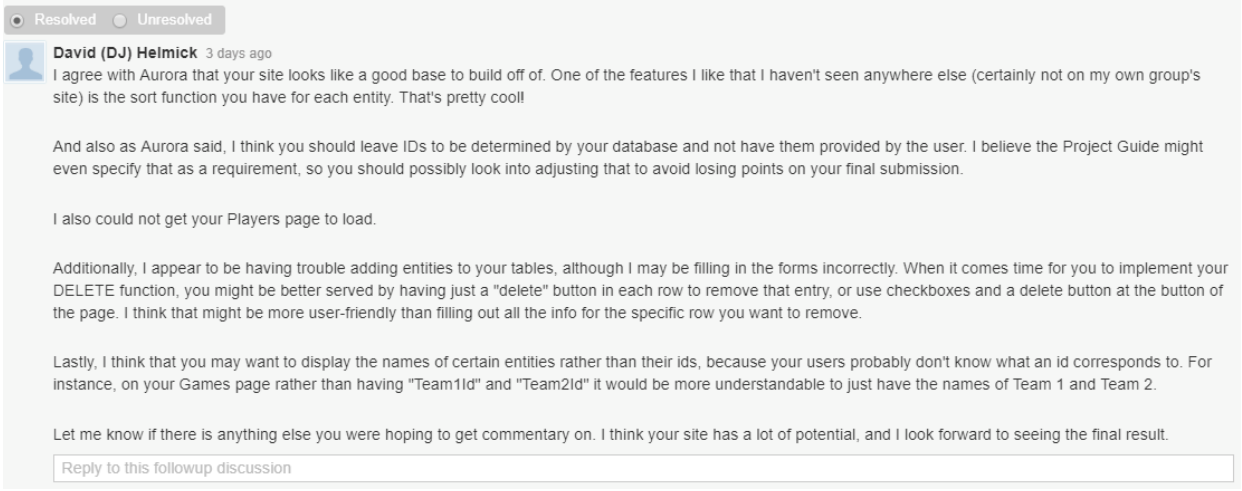
Gauge Hartwell
Allen Amusin
CS340 -
section 401

Project Step 5 Final Version: UPDATE Operations

https://web.engr.oregonstate.edu/~hartwelg/

Feedback by the reviewers:





Post1:

Aurora Weiden 4 days ago

This looks like a good start Allen, just make sure you start linking up functions to CREATE and READ for this step's final submission. I also have a few general comments -

Since ID's generally auto-increment I do not think users should be able to input an ID when they add a Team, Game, etc.

Additionally, on the first page (the index) the title is "Team 32 Basketball Database." Is the Team 32 a typo? Also, I think your Game page is missing the buttons to submit Game addition, deletion, update and info, and your score page is missing submit buttons to add and update.

And lastly the Players page is not loading properly on my end. I am not sure if that is only an issue for me, but I thought I would mention it in case this is a general problem.

Hope this information is helpful! Let me know if you would like me to elaborate on anything or if there is anything in particular you want me to review!

Post 2:
I agree with Aurora that your site looks like a good base to build off of. One of the features I like that I haven't seen anywhere else (certainly not on my own group's site) is the sort function you have for each entity. That's pretty cool!

And also as Aurora said, I think you should leave IDs to be determined by your database and not have them provided by the user. I believe the Project Guide might even specify that as a requirement, so you should possibly look into adjusting that to avoid losing points on your final submission.

I also could not get your Players page to load.

Additionally, I appear to be having trouble adding entities to your tables, although I may be filling in the forms incorrectly. When it comes time for you to implement your DELETE function, you might be better served by having just a "delete" button in each row to remove that entry, or use checkboxes and a delete button at the button of the page. I think that might be more user-friendly than filling out all the info for the specific row you want to remove.

Lastly, I think that you may want to display the names of certain entities rather than their ids, because your users probably don't know what an id corresponds to. For instance, on your Games page rather than having "Team1Id" and "Team2Id" it would be more understandable to just have the names of Team 1 and Team 2.

Let me know if there is anything else you were hoping to get commentary on. I think your site has a lot of potential, and I look forward to seeing the final result.

Actions based on the feedback:

- Updated index.html page to fix typo
- FIxed players page not loading error
- Added notes to each page to help user with adding to tables
- Fixed add and update operations that needed foreign key values -> turned them into select menus

a) Project Outline and Database Outline – Updated Version:

Project Outline

We will be creating a database representing the 2018 National Basketball Association (NBA). The NBA is a professional sports league consisting of 30 teams, each with a mascots and players. Teams are further broken down into 6 divisions each, with 5 teams in each division. Statistics are kept for each player to compare and track their abilities. To model this, we will use 4 entities: BasketballTeams, BasketballTeamPlayer, BasketballGame, and LeadingScore.

Database Outline

Entity: BasketballTeams		
Attribute	Data Type	Description
id	Integer, not null	Auto-incrementing integer automatically assigned when a row is created in this entity.
name	Varchar, not null	name of the team. Varchar value. No defaults, no empty strings
mascot	Varchar, not null	string containing the mascot name. Varchar value of up to 30 characters. No empty strings, no defaults.
NumWins	Integer, not null	number of wins in the current season. Int value. No empty values, no defaults.
NumLosses	Integer, not null	number of losses in the current season. Int value. No empty values, no defaults.
BasketballTeamId	Integer, not null	Id of the basketball team. Automatically assigned int value, automatically incrementing.

Entity: BasketballTeamPlayer		
Attribute	Data Type	Description
Id	Integer, not	Id of the player on the basketball team. Int

	null	value, automatically incrementing, automatically assigned, no empty values.
BasketballTeamId	Integer, not null	id of the team the basketball player belongs to. Int value, connects to BasketballTeamId in other tables.
FirstName	Varchar, not null	first name of the player. String value of up to 60 characters. No empty values, no default names.
LastName	Varchar, not null	last name of the player. String value of up to 60 characters. No empty values, no default names.
JerseyNumber	Integer, not null	number on the player's jersey. Int value, no empty strings, no default values.
Position	Varchar, not null	position played by the player. String value of up to 60 characters. No empty strings. Automatically assigned, defaults to either PG, SG, SF, PF, C.

Entity: BasketballGame		
Attribute	Data Type	Description
id	Integer, not null	id of the game. Int value. Automatically assigned, no empty strings, no defaults.
FinalScoreOfWinningTeam	Integer, not null	final score for the game for the winning team. Int value. No empty values,

		defaults to 0.
FinalScoreOfLosingTeam	Integer, not null	final score for the game for the losing team. Int value. No empty values, defaults to 0.
Location	Varchar, not null	location for the basketball game. String value of up to 60 characters. No empty strings, no defaults.
GameId	Integer, not null	id of the basketball game. Int value, automatically assigned, automatically incrementing, no empty values.
Team1Id	Integer, not null	id of team 1. Int value. Links to BasketballTeamId in other tables.
Team2Id	Integer, not null	id of team 2. Int value. Links to BasketballTeamId in other tables.

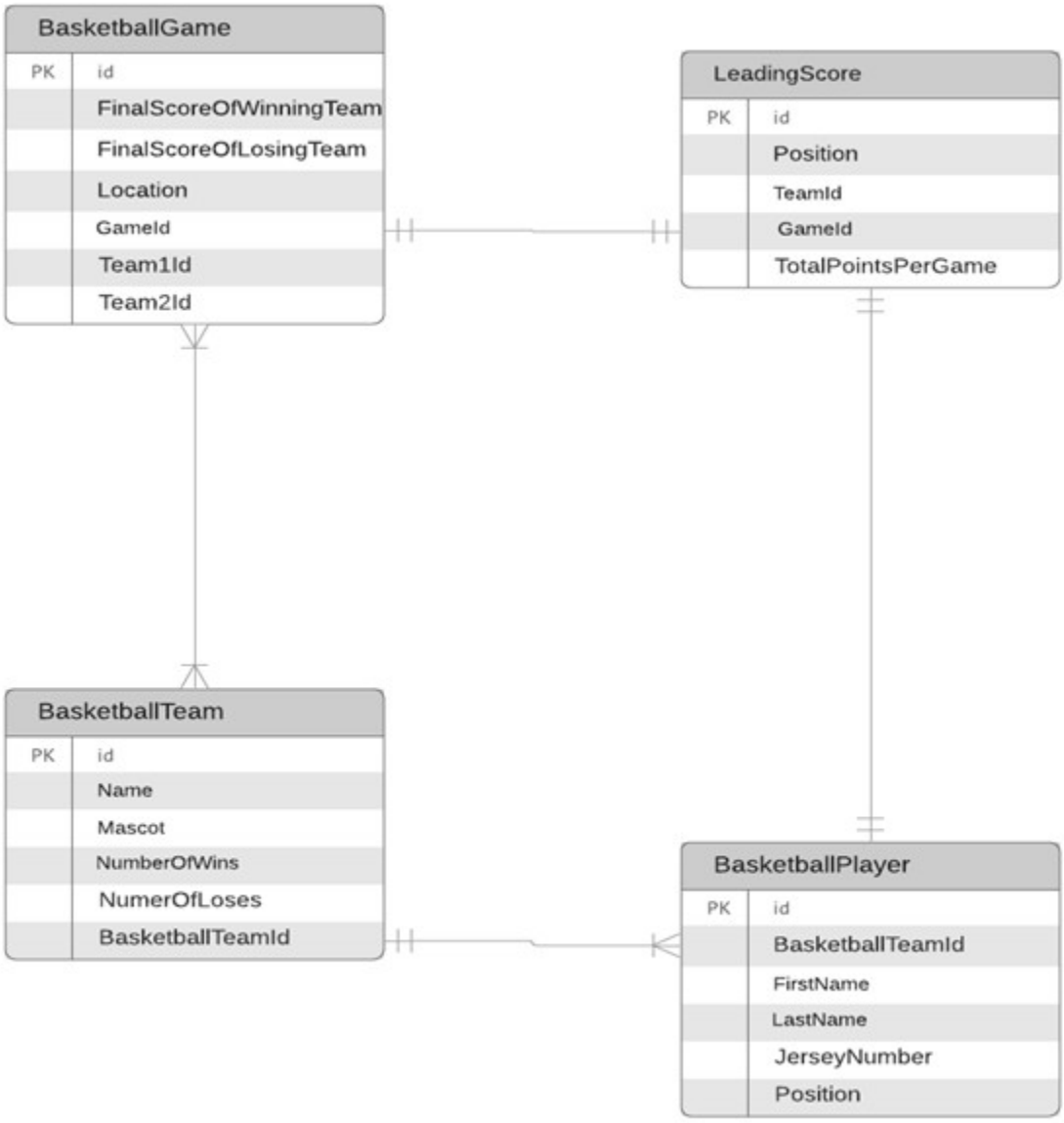
Entity: LeadingScore		
Attribute	Data Type	Description
id	Integer, not null	Id of the BasketballTeamPlayer with the leading score. Links to Id in the BasketballTeamPlayer table.
Position	Varchar, not null	Int of value 1 or 2, to indicate winning or losing team. No empty values.
TeamId	Integer, not null	Id of team with leading score. Int value, links to BasketballTeamId in other tables.
GameId	Integer, not null	Id of current game being played. Int value, links to GameId in other tables.
TotalPointsPerGame	Integer, not null	Total points of the BasketballTeam with the leading score. Int value, defaults to 0.

The relationships in our database are:

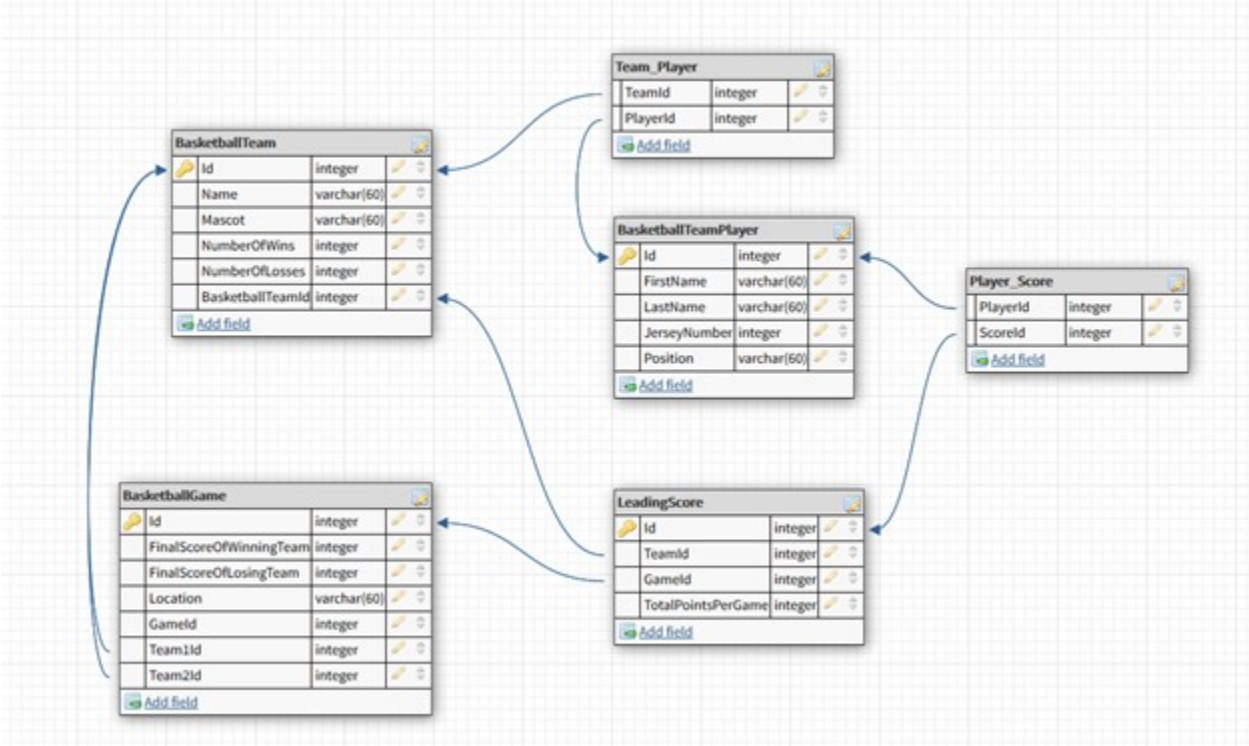
1. Many BasketballTeams can play many games, and BasketballGame can have many different teams play. Many to many relationship.
2. Each BasketballGame can have only one LeadingScore, and there is a LeadingScore in every BasketballGame. One to Many relationship.

3. Each BasketballTeamPlayer is part of a BasketballTeam, and a BasketballTeam can have many BasketballTeamPlayers. One to many relationship.
4. BasketballTeamPlayers can play in many BasketballGames, and each BasketballGame can have many players. Many to many relationship.

ERD


BasketballGame
PK
id
FinalScoreOfWinningTeam
FinalScoreOfLosingTeam
Location
GameId
Team1Id
Team2Id
LeadingScore
PK
id
Position
TeamId
GameId
TotalPointsPerGame
BasketballTeam
PK
id
Name
Mascot
NumberOfWins
NumerOfLoses
BasketballTeamId
BasketballPlayer
PK
id
BasketballTeamId
FirstName
LastName
JerseyNumber
Position

Schema



b) Fixes based on Feedback from Previous Steps:

TA Feedback:

We received a perfect score on our step 1, and we then got feedback on our original project casino database. We updated our project when turning in Project Step 2 final Draft Version: ERD & Schema and we did not get feedback on that part.

Peer Feedback:

we then got feedback on our original project casino database. We updated our project when turning in Project Step 2 final Draft Version: ERD & Schema and we did not get feedback on that part.

Here is the feedback we got for the casino database, but we can not use it since we changed the project.