Gauge Hartwell
Allen Amusin
CS340 -
section 401

Project Step 6 Final Version: DELETE Operations

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

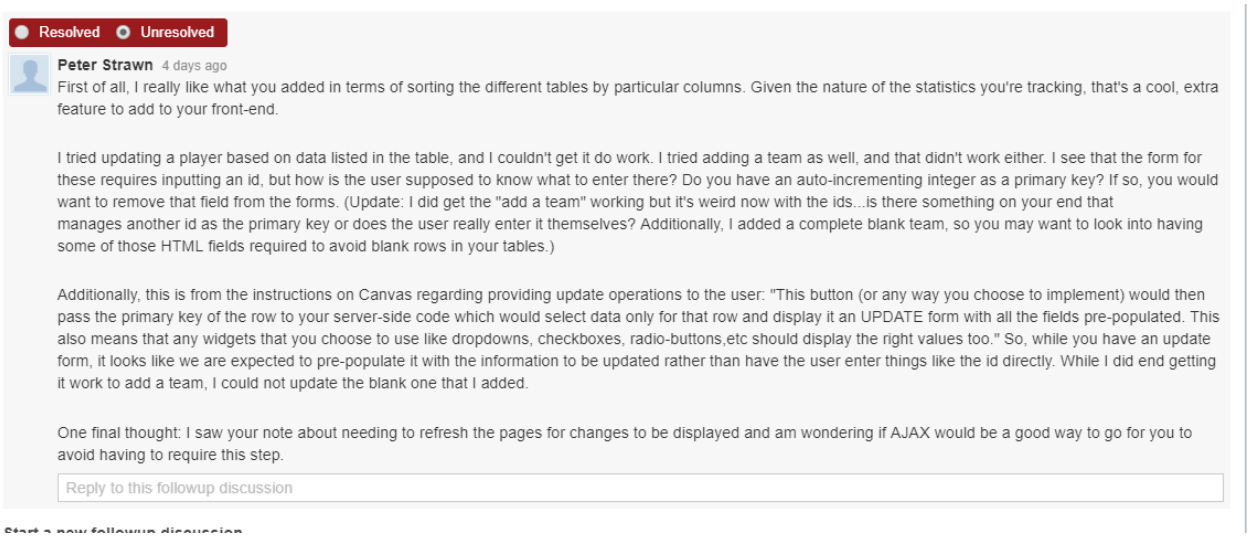
Feedback by the reviewers:

First of all, I really like what you added in terms of sorting the different tables by particular columns. Given the nature of the statistics you're tracking, that's a cool, extra feature to add to your front-end.

I tried updating a player based on data listed in the table, and I couldn't get it do work. I tried adding a team as well, and that didn't work either. I see that the form for these requires inputting an id, but how is the user supposed to know what to enter there? Do you have an auto-incrementing integer as a primary key? If so, you would want to remove that field from the forms. (Update: I did get the "add a team" working but it's weird now with the ids...is there something on your end that manages another id as the primary key or does the user really enter it themselves? Additionally, I added a complete blank team, so you may want to look into having some of those HTML fields required to avoid blank rows in your tables.)

Additionally, this is from the instructions on Canvas regarding providing update operations to the user: "This button (or any way you choose to implement) would then pass the primary key of the row to your server-side code which would select data only for that row and display it an UPDATE form with all the fields pre-populated. This also means that any widgets that you choose to use like dropdowns, checkboxes, radio-buttons,etc should display the right values too." So, while you have an update form, it looks like we are expected to pre-populate it with the information to be updated rather than have the user enter things like the id directly. While I did end getting it work to add a team, I could not update the blank one that I added.

One final thought: I saw your note about needing to refresh the pages for changes to be displayed and am wondering if AJAX would be a good way to go for you to avoid having to require this step.



Post1:
Yay!! I added a new team (Owls).
I completely understand about WIP... just wondering why the page needs to be reloaded in order for the new entered info to show up? Is this part of the stuff you guys are still working on?
Also, where are the Lakers? :-)

Post 2:
Hello,
Functionality wise, your project seems to be on a good pace. As for some things that I might suggest, I think that checking for valid input might be useful. For the add player function, have a check that makes sure the ID input isnt already taken. One idea might be to make the ID auto increment and make it so that the user does not have to give an ID since the system will automatically assign the ID when the user adds a new item. This way error checking wont be needed on the ID input. I also think that the delete player and update player only need the ID to delete or update the item since the ID seems to be unique/ the primary key. I also noticed that once you use the search functionality you cant go back to viewing a list of all the players/teams/etc without reloading, so a reset search button may be useful as well.

Actions based on the feedback:

- Did not add the Lakers to the database
- Fixed past issue where page did not update after data was entered/deleted/updated

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 null	Id of the player on the basketball team. Int 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.

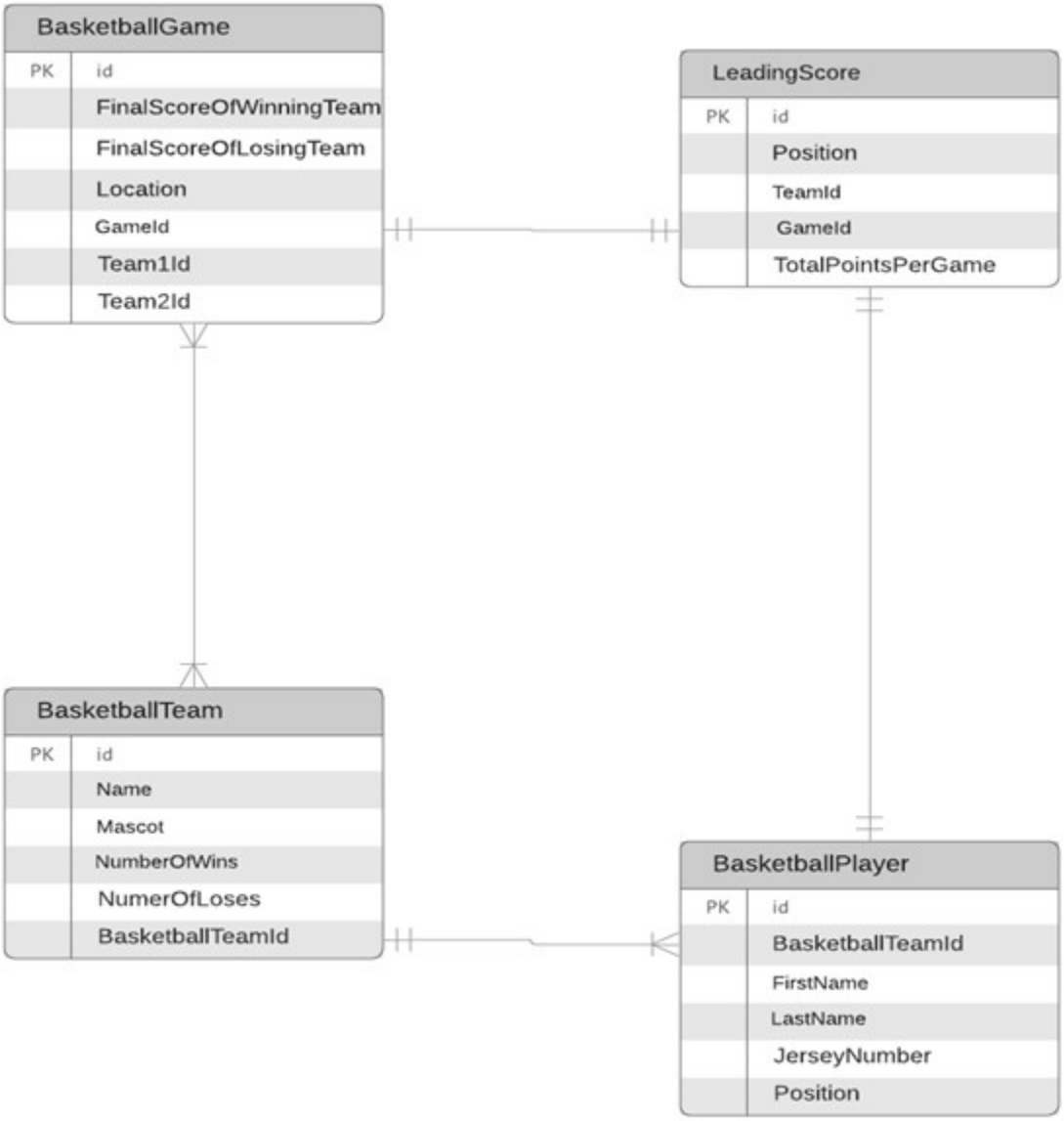
Entity: LeadingScore		
Attribute	Data Type	Description

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.

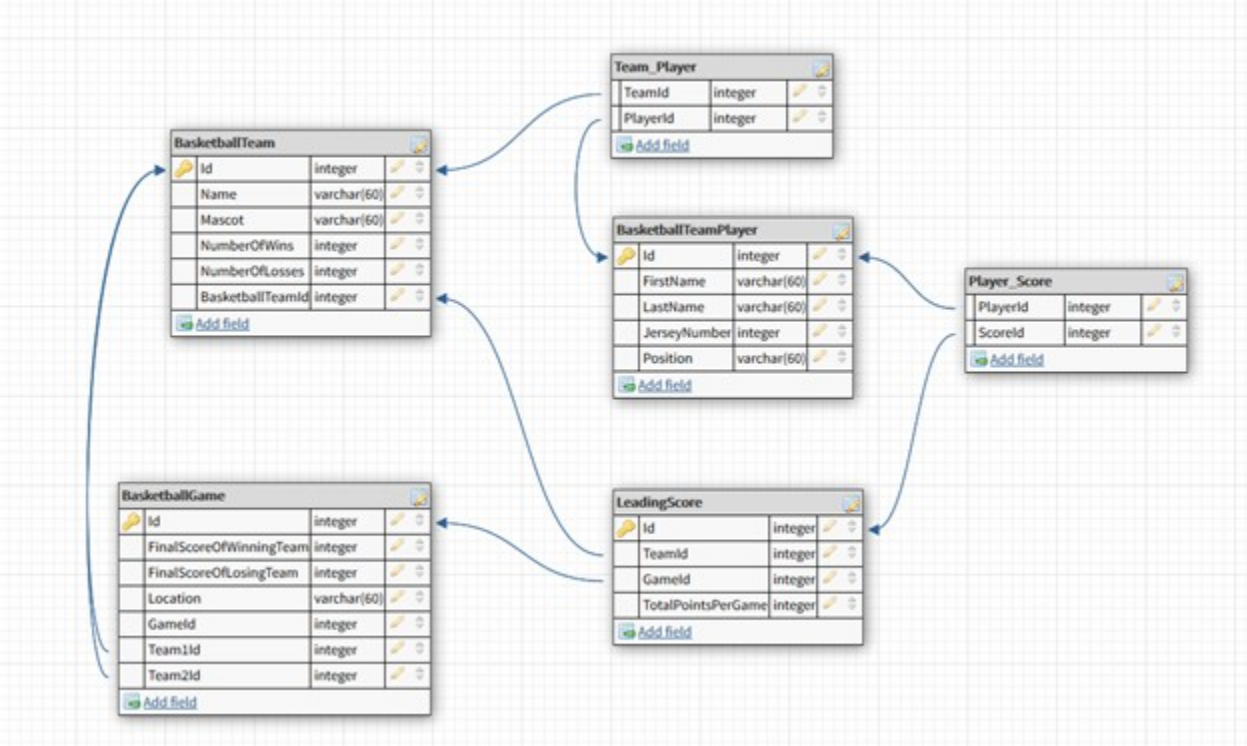
The relationships in our database are:

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.