HW 7 – Players: Editing with a Foreign Key

Contents

[1 Overview 1](#_Toc196736365)

[2 General Tips & Recommendations 1](#_Toc196736366)

[3 *editPlayers.php* – Initial Display 2](#_Toc196736367)

[4 *editPlayers.php* – Edit Mode 3](#_Toc196736368)

[5 *editPlayers.php* – Update & Cancel Modes 4](#_Toc196736369)

[6 *editPlayers.php* – Delete Mode 4](#_Toc196736370)

[7 Video Demo 5](#_Toc196736371)

[8 Submission Requirements 5](#_Toc196736372)

[Appendix 1 n/a 5](#_Toc196736373)

# Overview

You will create an app consisting of a single page, *editPlayers.php* that allows a user to edit and delete players from the *league* database. In the next few sections, we’ll step through the requirements the page and provide tips/hints.

# General Tips & Recommendations

After coding this myself, I note the following:

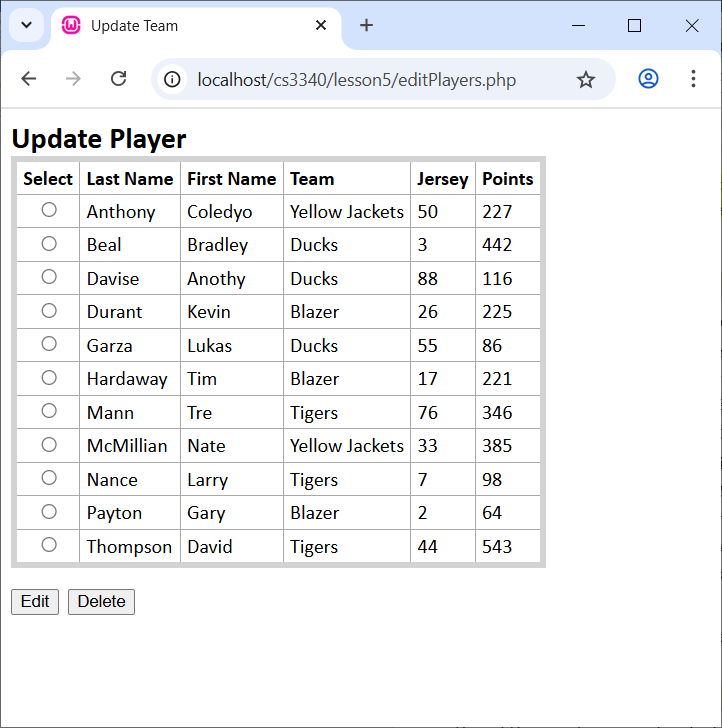
* I recommend that you use methods for much of the work as possible. In my solution, I strived to write the shortest <body> for each of the pages, trying to show the logic that was taking place about which parts/actions needed to be displayed, and then using methods to do display.
* I grouped my methods into 3 categories and stored them in external files:

1. *sql\_util.php* – Contains methods that do SQL operations: *build\_connection, retrieveTeams, retrievePlayers, retrievePlayer, updatePlayer, deletePlayer*
2. *build\_util.php –* Contains methods that build and return display objects: *buildPlayersTableHeaderRow, buildPlayerRow, buildTeamDropDown*.
3. *display\_util.*php – Contains methods to display objects: *displayPlayersTable, displayPlayerToEdit*.

* Use baby-steps!
* *Var\_dump(object)* is your best friend. It shows everything about an object: class, instance variables, data types, values, *etc.*

# *editPlayers.php* – Initial Display

When the page is first loaded it appears as shown below



**Hints/Suggestions:**

1. Similar to HW 6, the radio buttons in the table need to have the *value* attribute set to the *PlayerID* so that when *Edit* (*Delete*)is pressed and the page posts back, you’ll be able to know which player was selected for edit (deletion).
2. You’ll need an inner join to grab the players data and the team name. Remember that the *players* table has a foreign key, *TeamID,* which is used in the SQL statement to get the corresponding team name from the *teams* table.

$conn = build\_connection();  
$sql = "SELECT players.PlayerID, players.TeamID, teams.Name,

players.LastName, players.FirstName, players.Jersey, players.Points

FROM players

INNER JOIN teams ON teams.TeamID=players.TeamID

ORDER BY LastName, FirstName";

$stmt = $conn->query($sql);$stmt->execute();  
$players = $stmt->fetchAll(PDO::*FETCH\_ASSOC*);

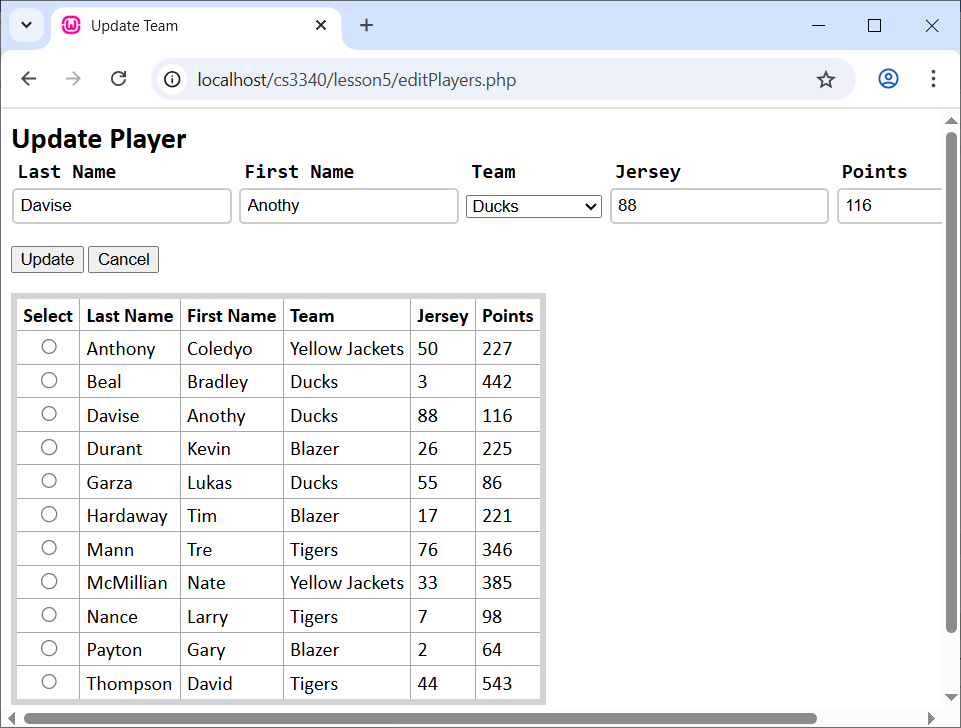
1. Note that the data is being returned in an array, *$players* where each element in the array is an associative array. For example, to iterate over the *$players* array:

foreach($players as $player) {  
 echo $player["PlayerID"] . ", " . $player["TeamID"] . ", " .

$player["LastName"];  
}

# *editPlayers.php* – Edit Mode

When a player is selected and the *Edit* button is pressed, the display shows text fields and a dropdown for the user to edit player data. The display also has *Update* and *Cancel* buttons, which are discussed in following sections.



**Hints/Suggestions:**

1. When *Edit* is pressed, it posts back the radio button that is selected. From that, you can obtain the *Value* which should contain the *PlayerID*. Which is used in the SQL statement to *SELECT* that player from the database using a *WHERE* clause. This is very similar to what you did in HW 6; however, here, I fetched into an associative array:

$stmt = $conn->prepare("SELECT \* FROM players WHERE PlayerID=?");  
$playerID = (int)$\_POST["playerID"];  
$stmt->execute([$playerID]);  
$player = $stmt->fetch(PDO::*FETCH\_ASSOC*);

1. **Very Important**. When you press *Update* (considered in the next section), you need to know what player you are editing, *i.e.* you need to know the *PlayerID*. Thus, you’ll need to stuff the *PlayerID* into a *hidden field* in the page. You don’t want to display it, but you do want it to post back. I stuck a line like this in a table row, but outside a *<td>* tag. Note the braces surrounding: {$player['PlayerID']} in the statement. This is different than in HW 6 because of the use of the associative array.

<input type="hidden" name="playerID" value="{$player['PlayerID']}">

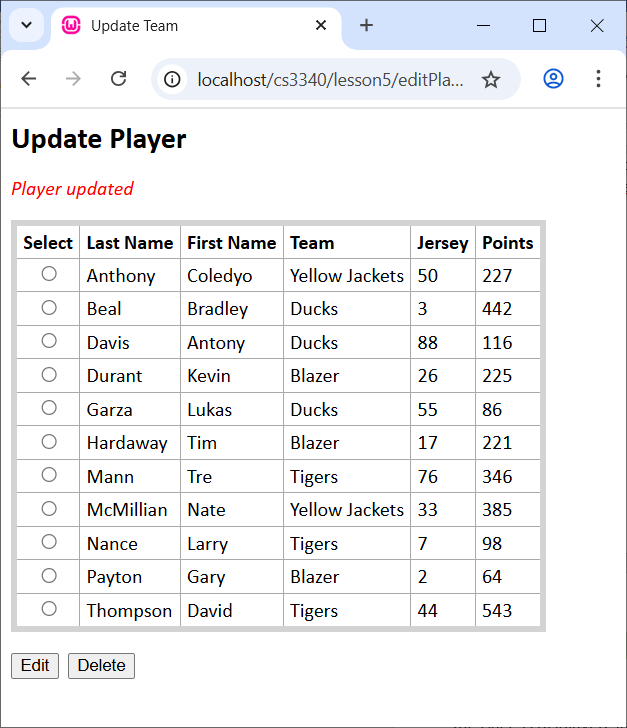
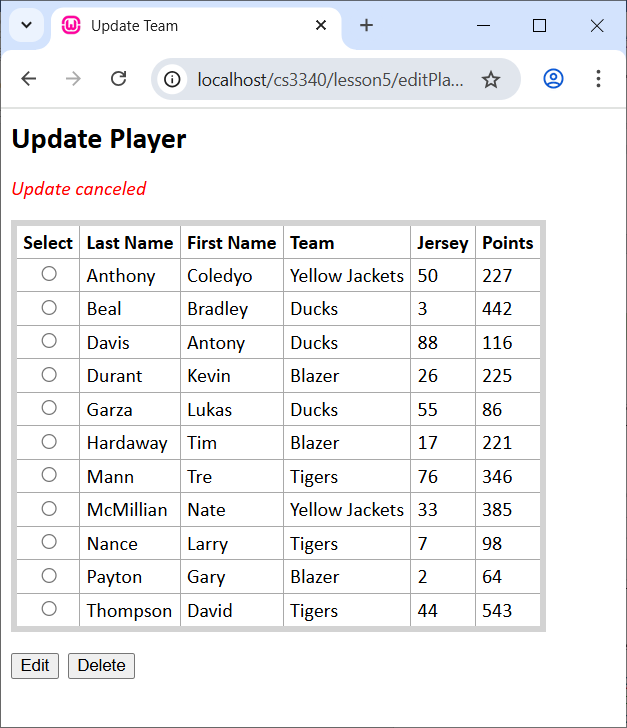
1. For the drop down, you’ll need another SQL statement to select all the teams. Note that the data is being returned in an array, *$teams* where each element in the array is an associative array. This is similar to Section 3, item 3 above.

$stmt = $conn->query("SELECT TeamID, Name FROM teams");$stmt->execute();  
$teams = $stmt->fetchAll(PDO::*FETCH\_ASSOC*);

1. When you build the drop down, you’ll want the *TeamID* in the *value* of he *option*. Making sure the correct team is displayed (*e.g.* the “Ducks” is displayed because that is the team “Anothy Davise” is currently on) is a bit trickier. You need to set the *selected* attribute for the *option* that corresponds to the team the player is currently on. Hint: you have the *TeamID* of the player being edited from the SQL statement in item 1 above.

# *editPlayers.php* – Update & Cancel Modes

When *Update* is pressed, and postback occurs, you want to: (a) Issue the SQL *UPDATE* statement, (b) Redisplay the page without the edit area, (c) Display a message confirming the update. When *Cancel* is pressed, a postback occurs and a message is displayed stating that the update was canceled.

**Hints/Suggestions:**

1. To update a team, I used code as shown below

$sql = "UPDATE players SET TeamID=?, LastName=?, FirstName=?, Jersey=?, Points=? WHERE PlayerID=?";  
$stmt = $conn->prepare($sql);

...  
$stmt->execute([$teamID, $lastName, $firstName, $jersey, $points, $playerID]);

1. Remember that the *PlayerID* and *TeamID* need to be integers.

# *editPlayers.php* – Delete Mode

When *Delete* is pressed, and postback occurs, you want to issue the SQL *DELETE* statement and display a confirmation message. This will be very similar to HW 6. Remember that players can always be deleted because *PlayerID* does not appear as a foreign key in any other tables. Make sure when you turn in there are at least 3 players on each of 4 teams (you can have more teams and players). You can issue a statement like this in PHPMyAdmin to add players (or you can do it manually:

INSERT INTO `players` (`PlayerID`, `TeamID`, `LastName`, `FirstName`, `Jersey`, `Points`) VALUES  
(1, 3, 'McMillian', 'Nate', 33, 385),  
(2, 5, 'Durant', 'Kevin', 26, 225),  
(3, 4, 'Thompson', 'David', 44, 543),  
(4, 3, 'Anthony', 'Cole', 50, 227),  
(5, 2, 'Beal', 'Bradley', 3, 442),  
(6, 2, 'Garza', 'Lukane', 55, 86),  
(7, 4, 'Mann', 'Tre', 76, 346),  
(8, 2, 'Davis', 'Anothy', 88, 116),  
(9, 4, 'Nance', 'Larry', 7, 98),  
(10, 5, 'Payton', 'Gary', 2, 64),  
(11, 5, 'Hardaway', 'Tim', 17, 221),  
(12, 3, 'Carter', 'Devin', 8, 664);

# Video Demo

Make a video demoing your app using Kaltura Capture ([Directions](https://knowledge.kaltura.com/help/d2l-add-media-kaltura-capture-overview)) in Blazeview. Do the following in video:

1. Open myPhpAdmin and display your database.
   1. Open your *players* table and display the data
   2. Open your *teams* table and display the data
   3. Select the database and open the Designer
2. Display *editPlayers.php:*
   1. Select a player to edit, indicate their current team in the table, and choose: Edit.
   2. When the edit area appears, indicate that the team being displayed in the drop down is the team that they are on.
   3. Change several things including changing the team the player is on and then choose: Update
   4. Indicate in the table that the player has been changed.
   5. Select a player to edit, then choose: Cancel.
   6. Select a player and choose: Delete.

# Submission Requirements

1. Zip the following files into a file named: *hw7\_yourLastName.zip*:
2. *editPlayers.php,*
3. any other files you created to support your app, including a *css* file.
4. Login into Blazeview, choose HW 7 assignment.
   1. Add the *hw7\_yourLastName.zip* file
   2. In the Comments area, choose: “Insert Stuff”, and then choose: Embed Kaltura Video and choose your video.

Appendix

1. n/a