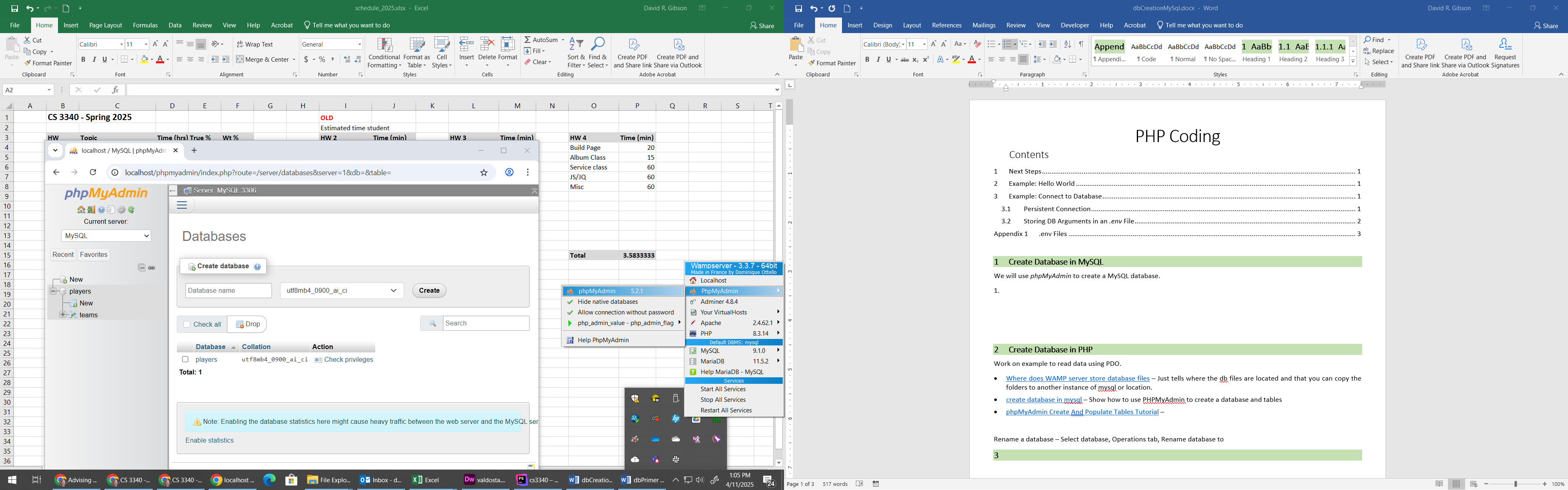
Creating a MySQL Database in phpMyAdmin

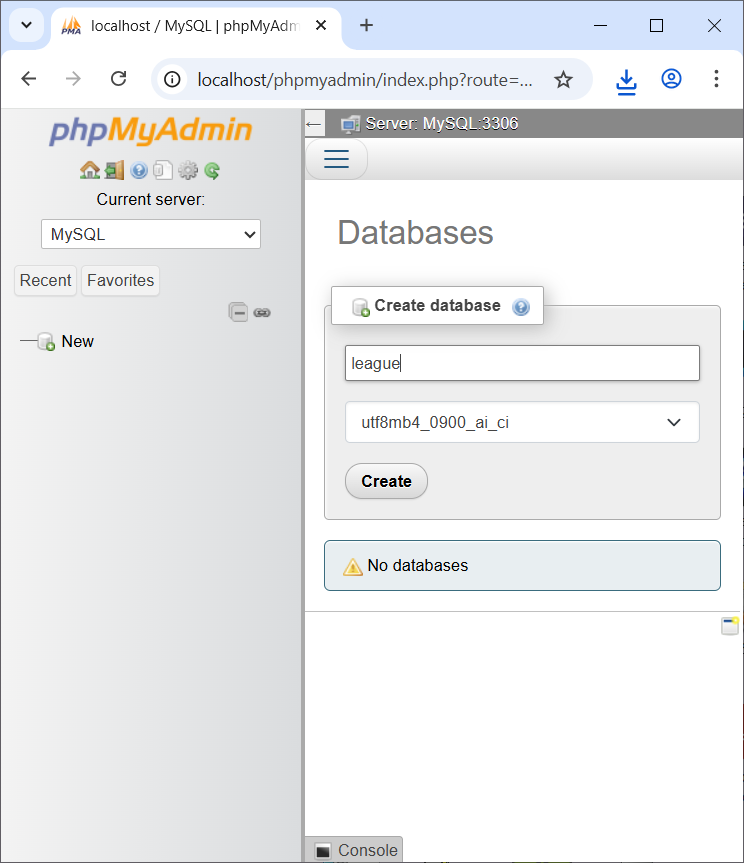
This document illustrates how to use *phpMyAdmin* to build a MySQL database, *league* with two joined tables: *teams* and *players.* We will mostly use a GUI interface; however, everything you do from the creation of the database to entering data into tables results in a SQL command which you can view or modify.

1. Start Wamp server and then open PhpMyAdmin

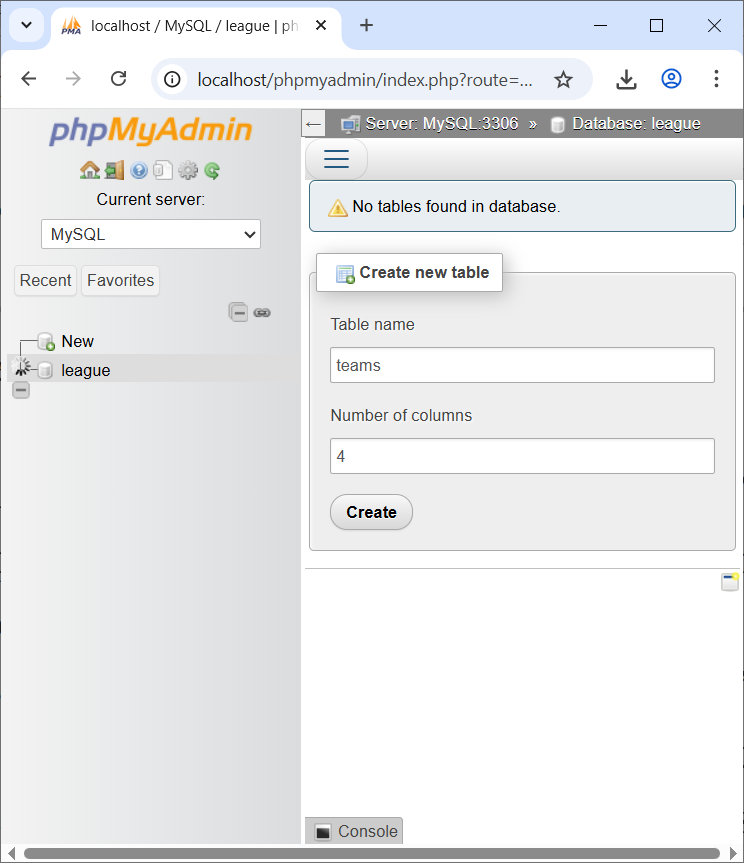


1. You’ll be asked to login. “root” is the default Username and leave the password blank.
2. Type, “league” as the name of the database and choose: Create.

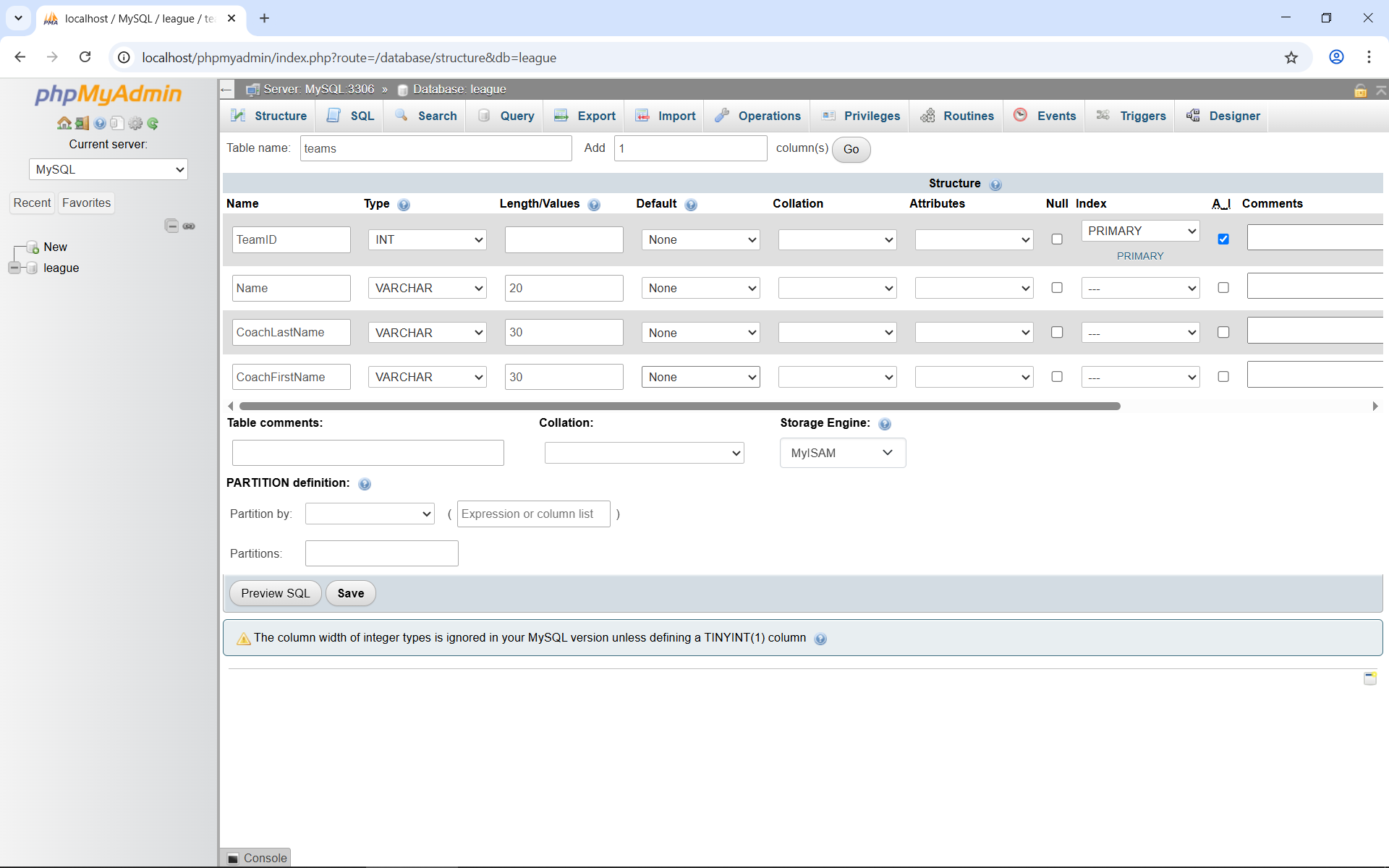
Note that you can also select “New” on the left and the same dialog will show.



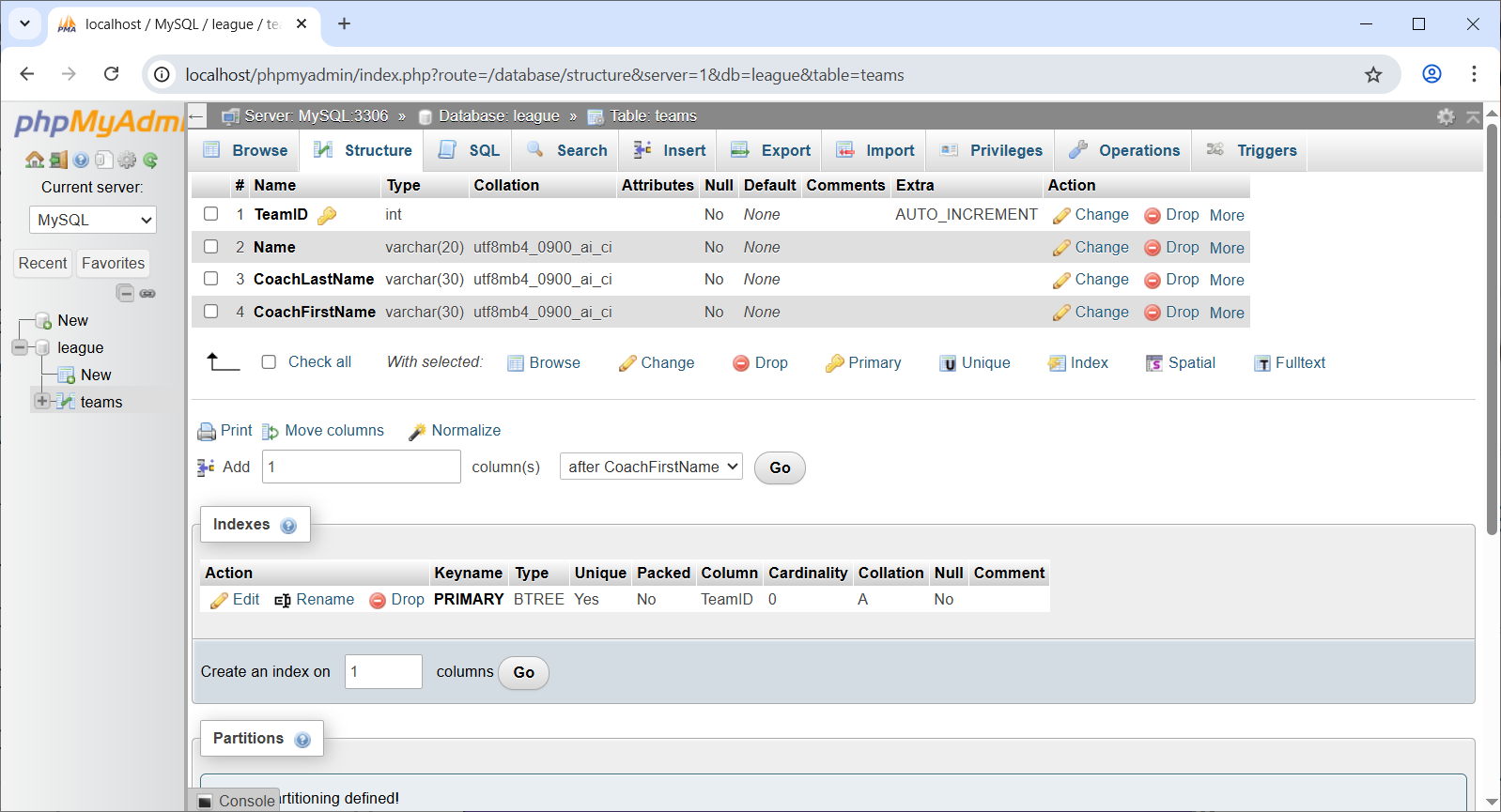
1. Type, “teams” for the table name and “4” for number of columns



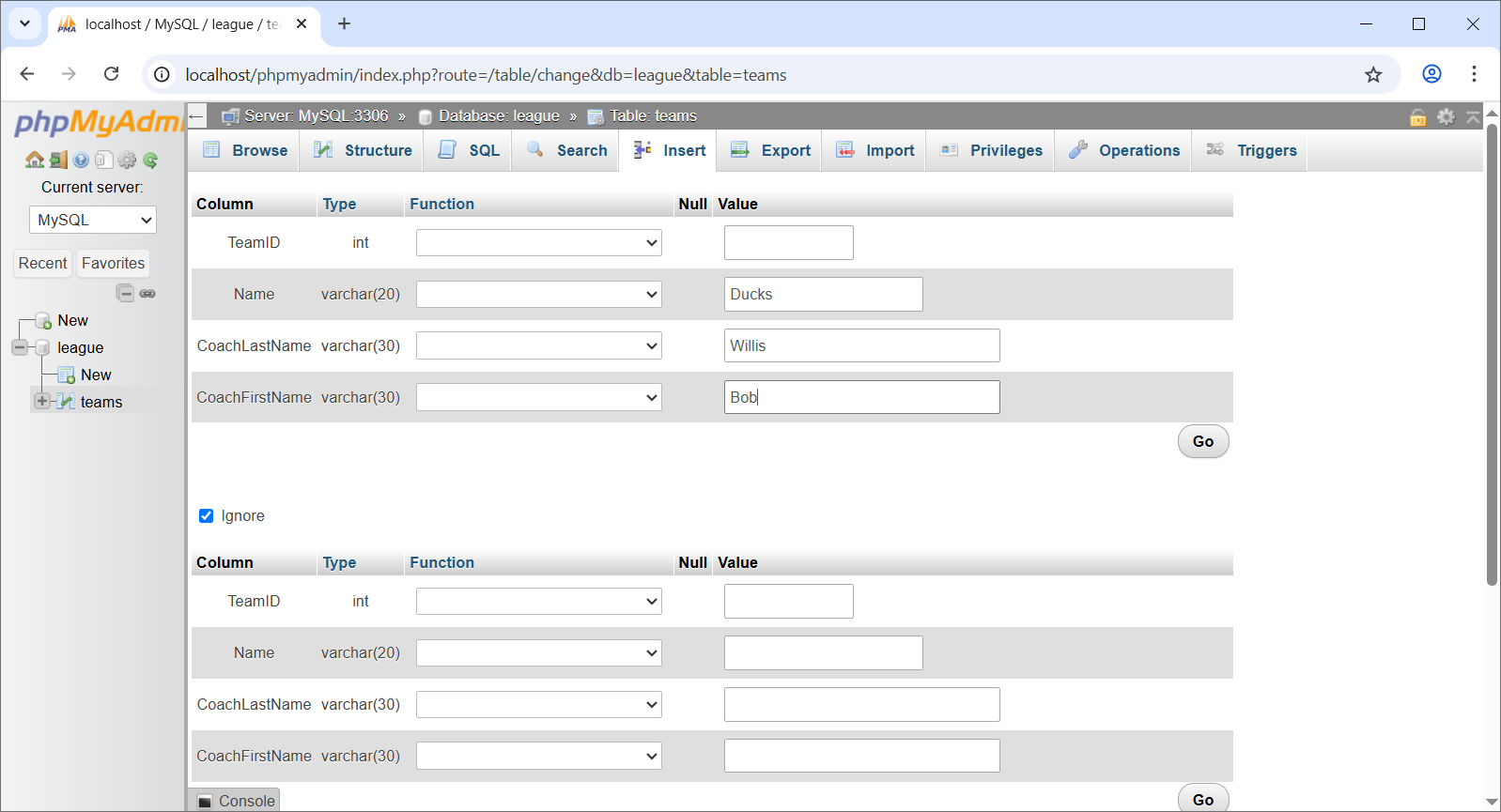
1. Supply the 4 fields (columns) exactly as shown below. Note that TeamID has “Index” set to “Primary” and that “A\_I” is checked. “A\_I” means “auto-increment”, which tells the db engine to be responsible for supplying the unique TeamID when a record is inserted. Choose: Save (not shown in image, it is below what is shown)



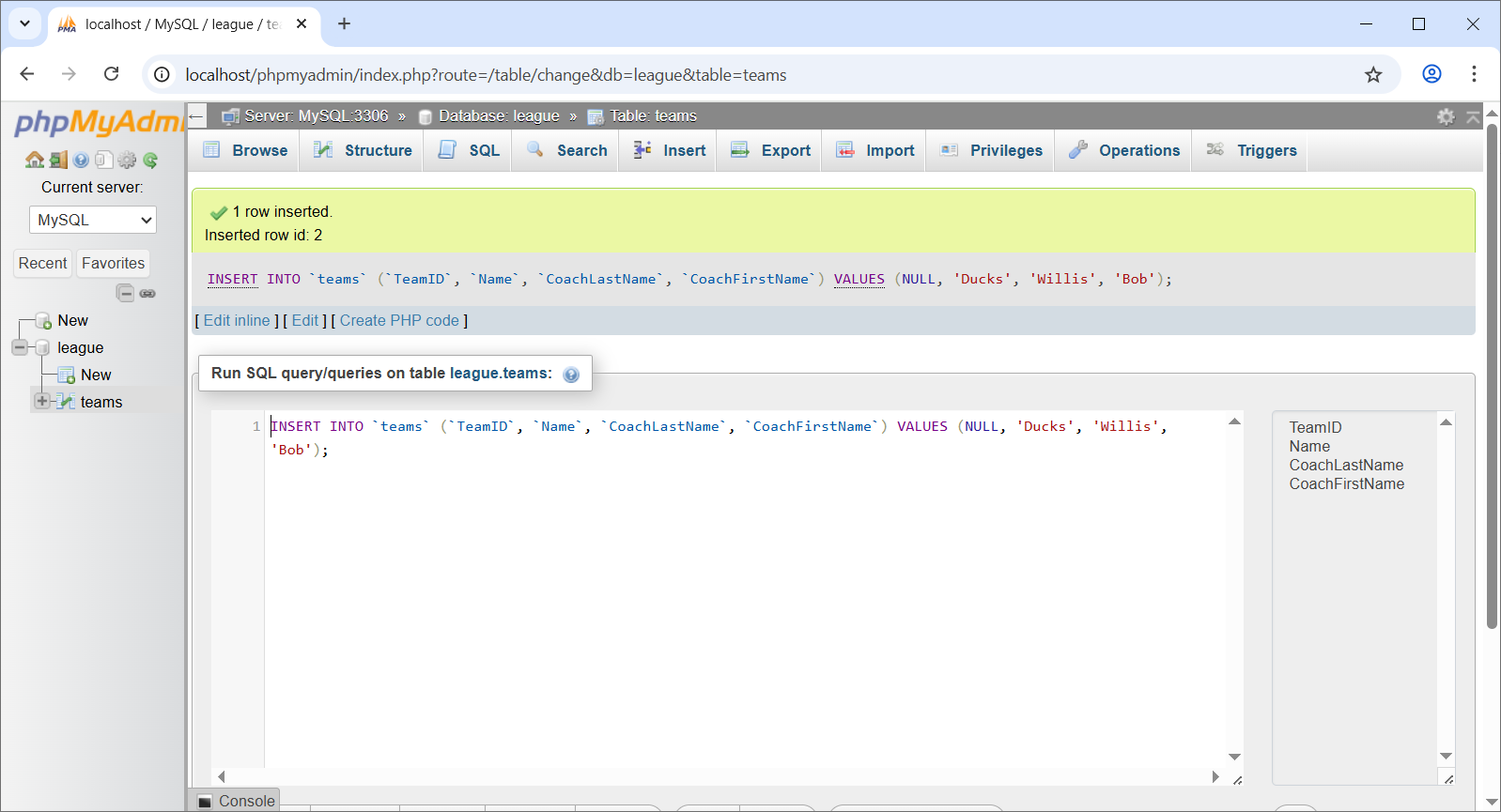
1. The result will look like as shown below. Note that the “Structure” tab is selected.



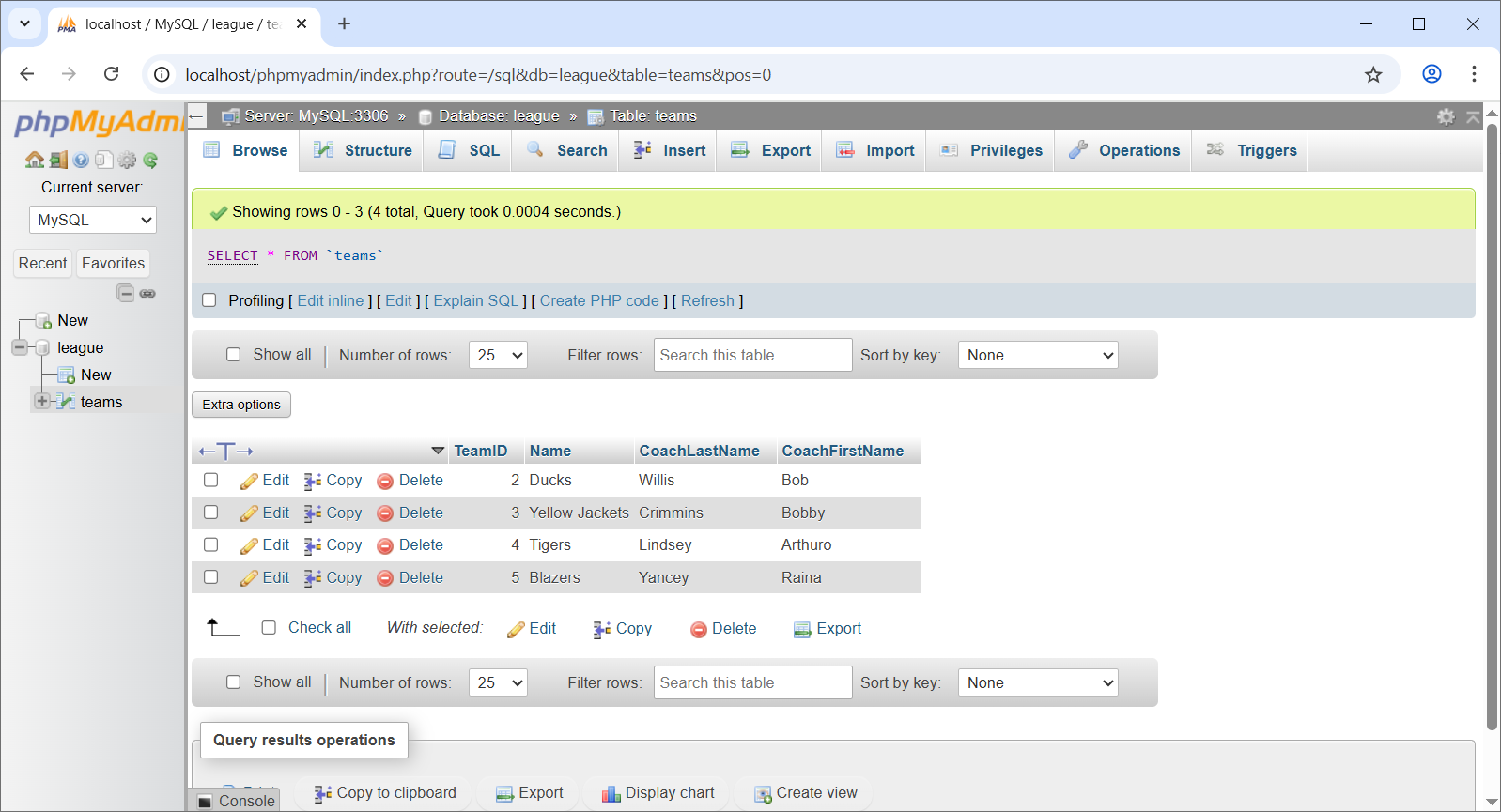
1. Select the “Insert” tab and type some data in (anything you want) and choose: Go. **DO NOT supply a TeamID.**



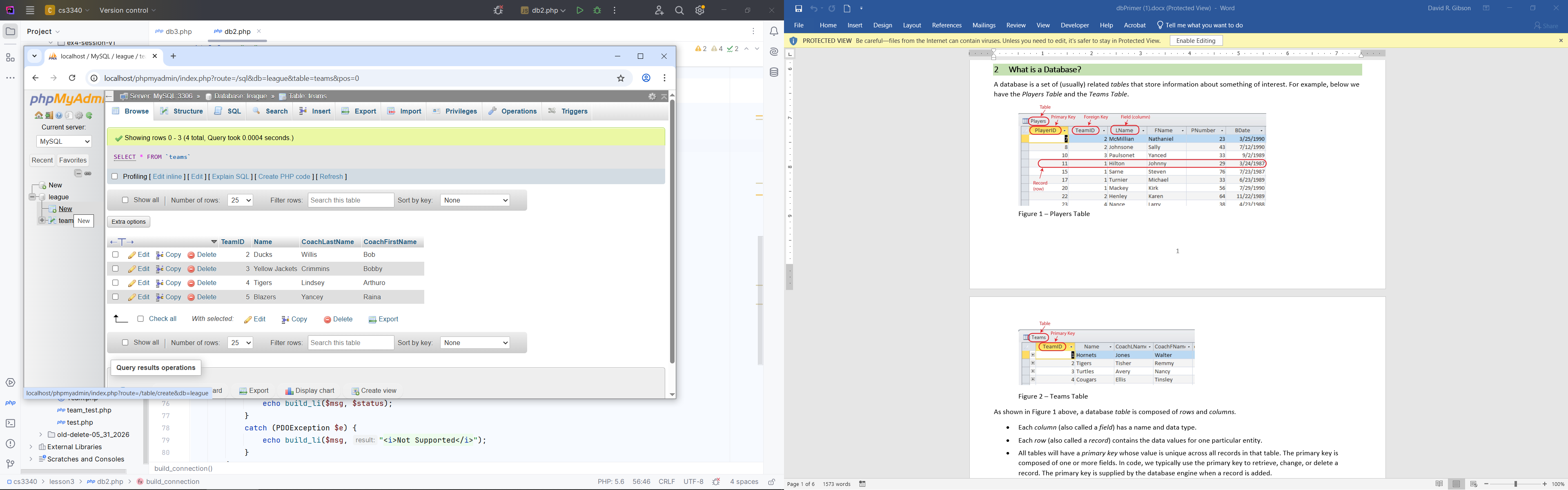
1. This will be shown:



1. Select the “Insert” tab and add another team. Do this until you have at least 4 teams.
2. Select the “Browse” tab and you will see your records. Note the TeamID’s. You’ll need them shortly. In my case, they are: 2,3,4,5. Yours will probably be: 1,2,3,4.

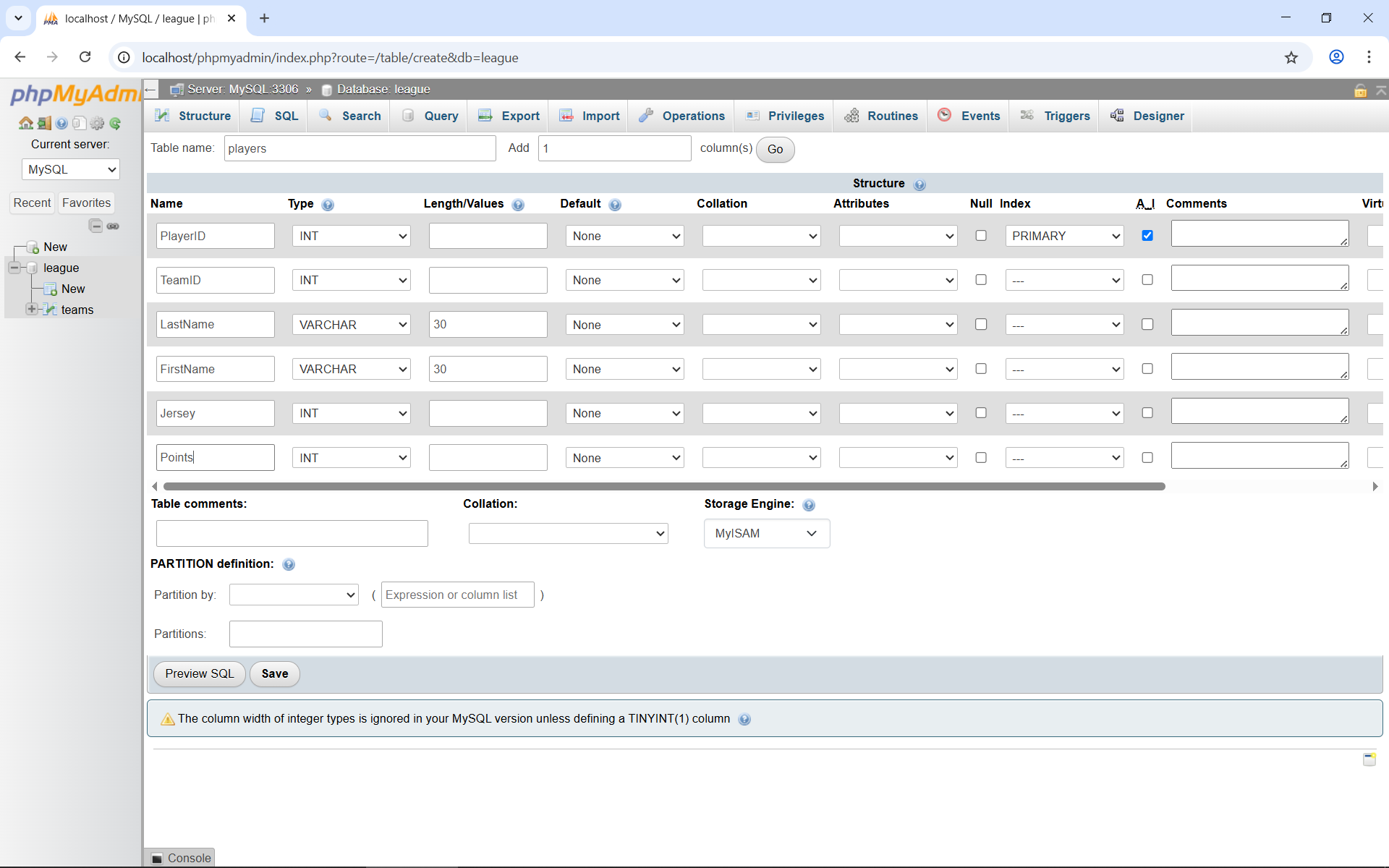


1. Next, we’ll create another table. On the left in the figure below, choose, “New” right above “team”.

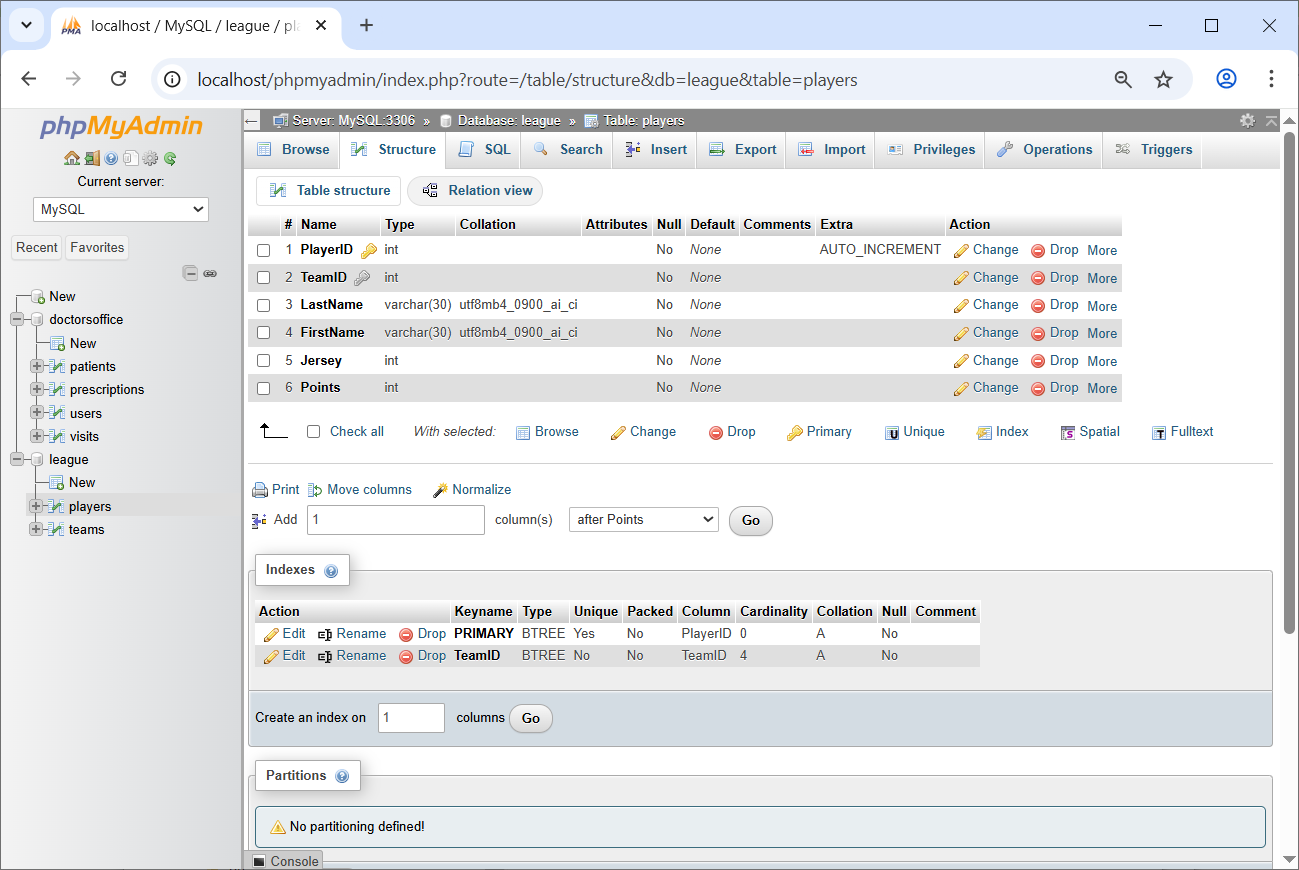


1. Type the name of the table, “players”. It will probably show 4 or so rows (for specifying the fields). You want 6 total, so supply the appropriate value in the “Add” textfield and choose: Go. Then, type the values below and choose: Save (not shown in figure, but is below what is shown). Note that the *PlayerID* is “Primary” and has “A\_I” checked.

**IMPORTANT: In the *TeamID* row, set the *Index* column to “Index”. This is NOT shown in the figure below.**



1. The result will look like as shown below. Note that the “Structure” tab is selected at the top and the *players* table is selected from left (not shown in figure). Note the indexes at the bottom. “Indexing is a way of sorting a number of records on multiple fields. Creating an index on a field in a table creates another data structure which holds the field value, and a pointer to the record it relates to.”[[1]](#footnote-1)



1. Make sure the “players” table is selected on the left and then choose: SQL tab and then INSERT (towards bottom). Type or paste this statement in and choose: GO in the lower-ish right. Note that the “2” below is the TeamID. That is the first team in my table, yours will probably be 1. Choose: Browse to see the records.

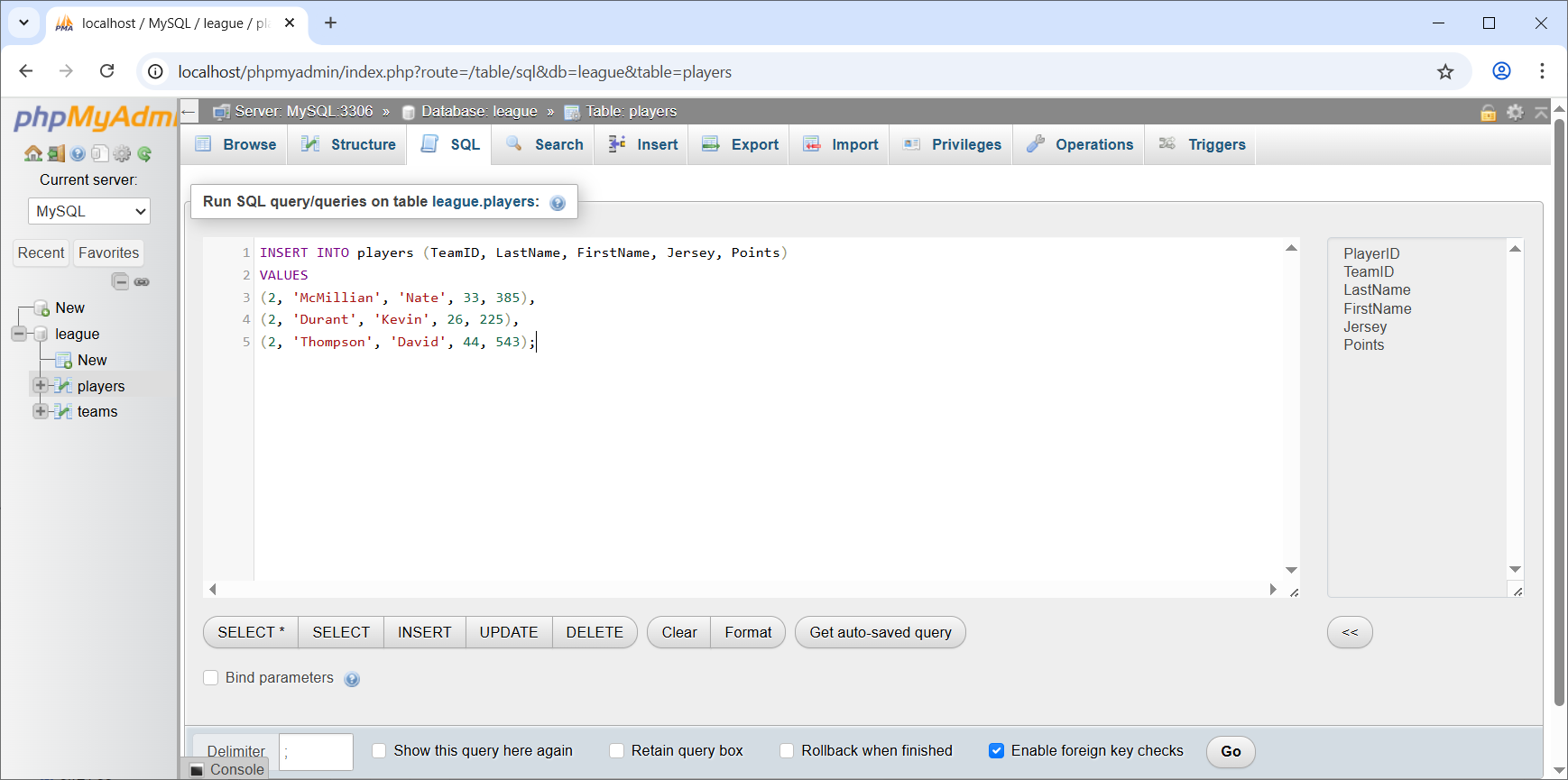
INSERT INTO players (TeamID, LastName, FirstName, Jersey, Points)

VALUES

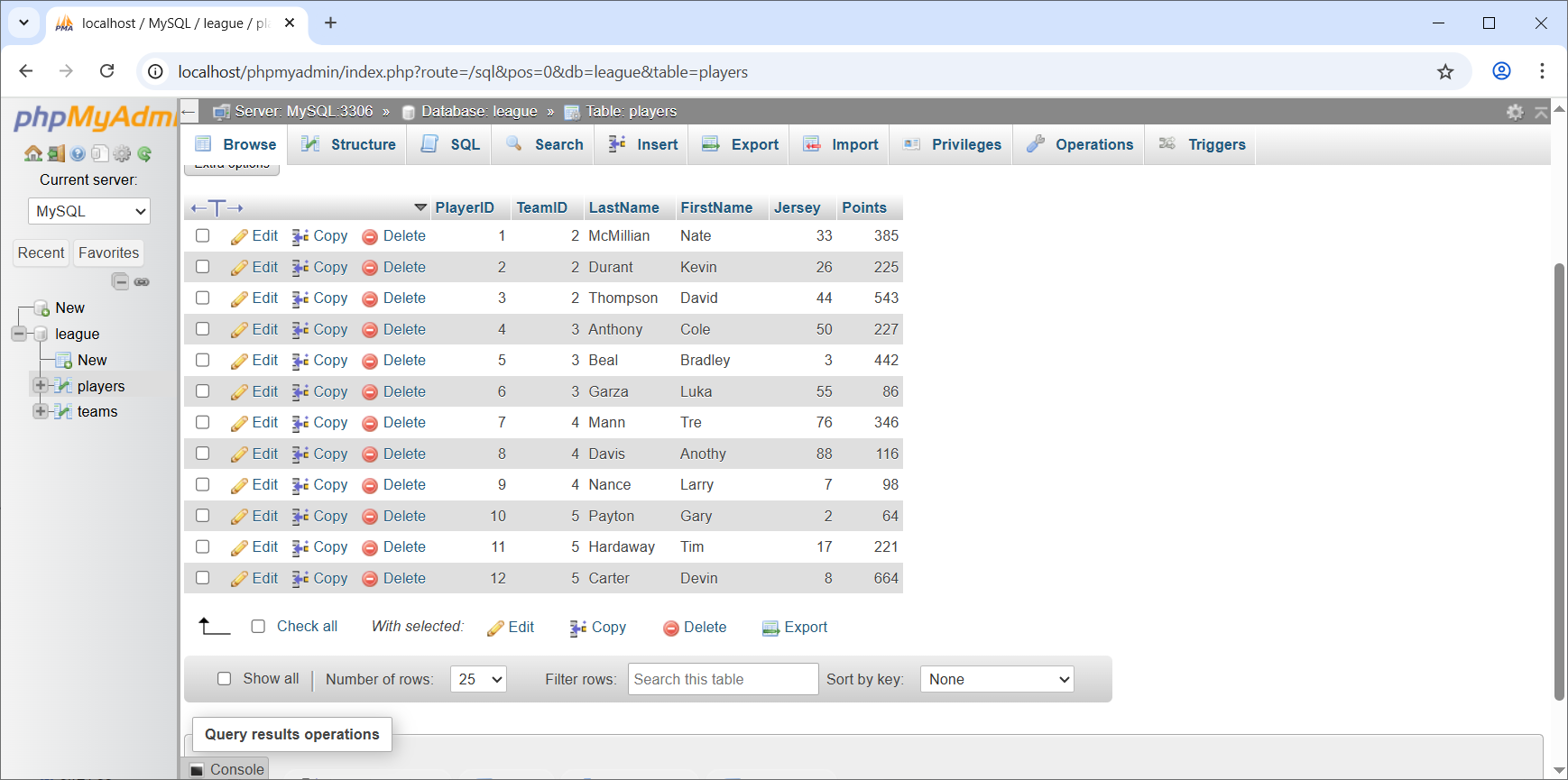
(2, 'McMillian', 'Name', 33, 385),

(2, 'Durant', 'Kevin', 26, 225),

(2, 'Thompson', 'David', 44, 543);



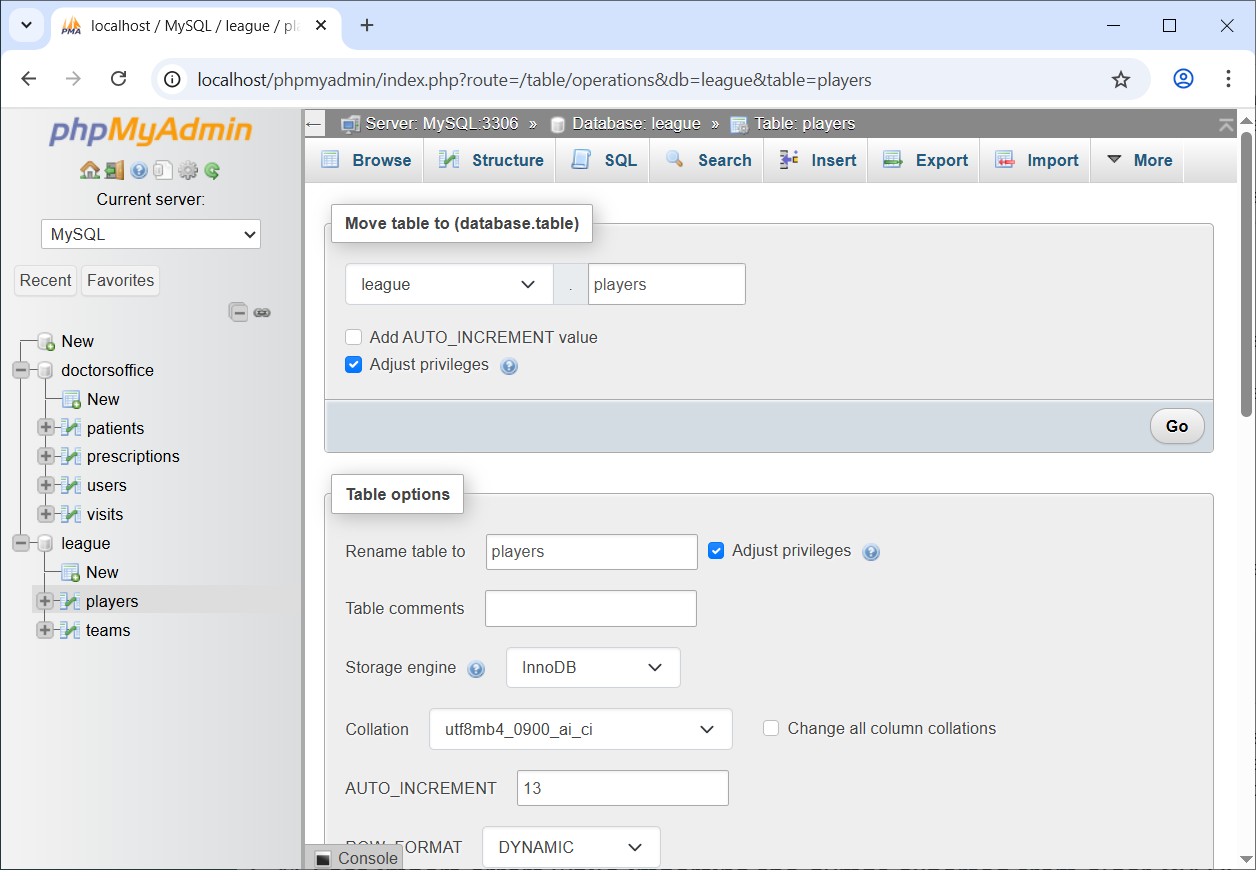
1. Repeat this so that you have at least 3 players on each team. You can add them all with one insert. At the end, you should have 12 or more:



1. Next, we need to explicitly define the primary key – foreign key relationship between the *teams* and *players* tables. To do this, we will use the *Designer* tool and to use the *Designer* tool, we must first change the “Storage Engine”! Follow the steps below to do this. The source for this is below. However, there, the source is enabling the *Relation* view. The *Relation* view is another way you can define the PK-FK relationship.

Source: <https://manage.accuwebhosting.com/knowledgebase/2567/How-to-Enable-Relation-View-in-phpMyAdmin.html>

* 1. Select the *players* table and choose: Operations.
  2. Under “Table options”, set the “Storage engine” to: “innoDB” and then choose: Go.

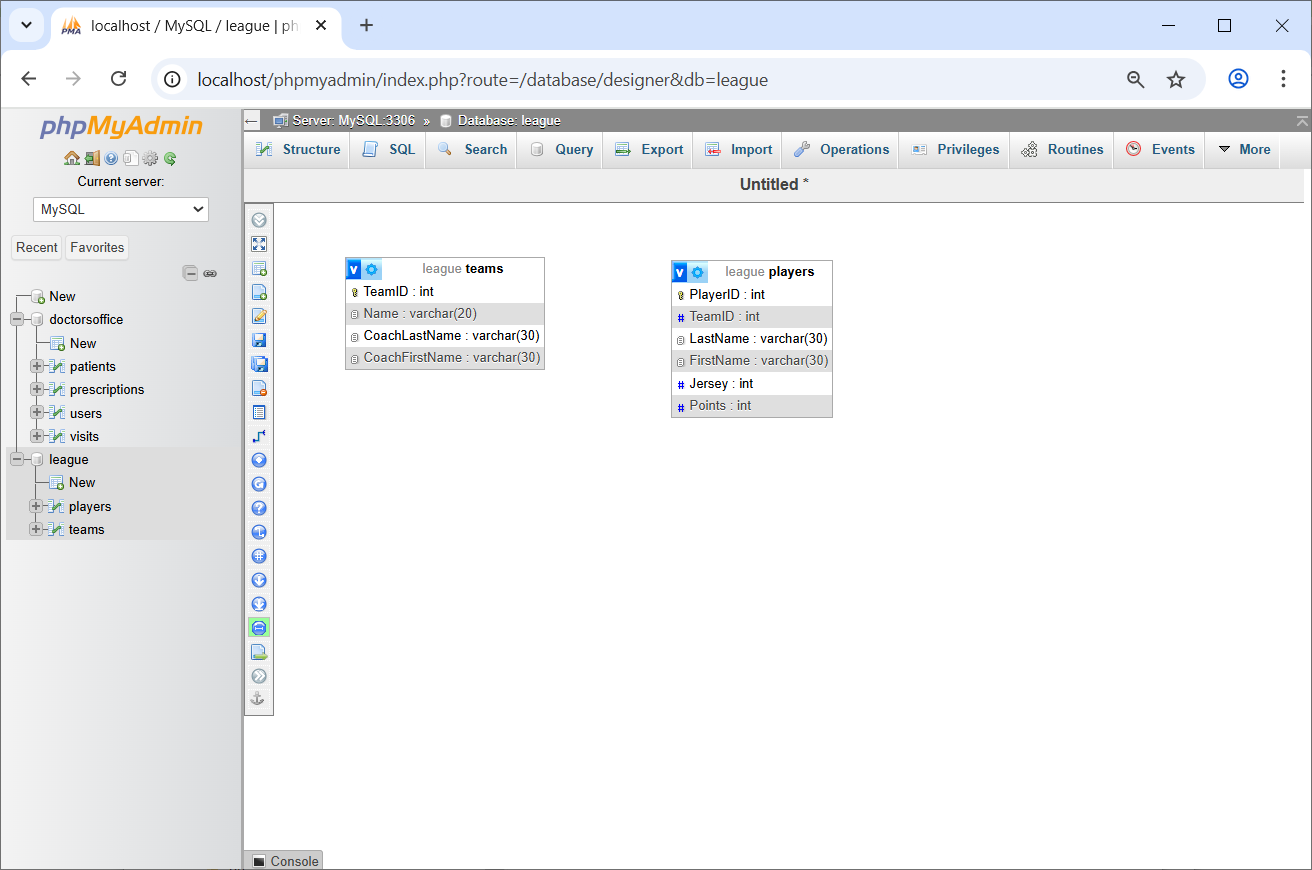


* 1. **Do the same thing for the *teams* table.**

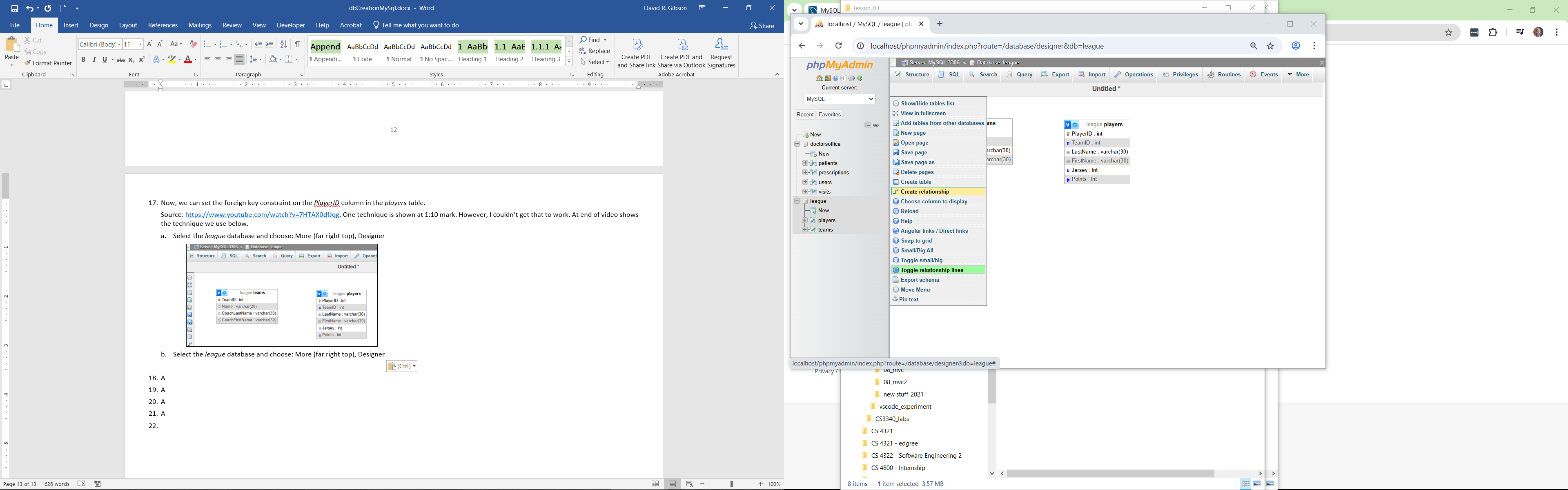
1. Now, we can set the foreign key constraint on the *PlayerID* column in the *players* table.

Source: <https://www.youtube.com/watch?v=7HTAX0dfJqg>. One technique is shown at 1:10 mark. However, I couldn’t get that to work. At end of video itshows the technique we use below.

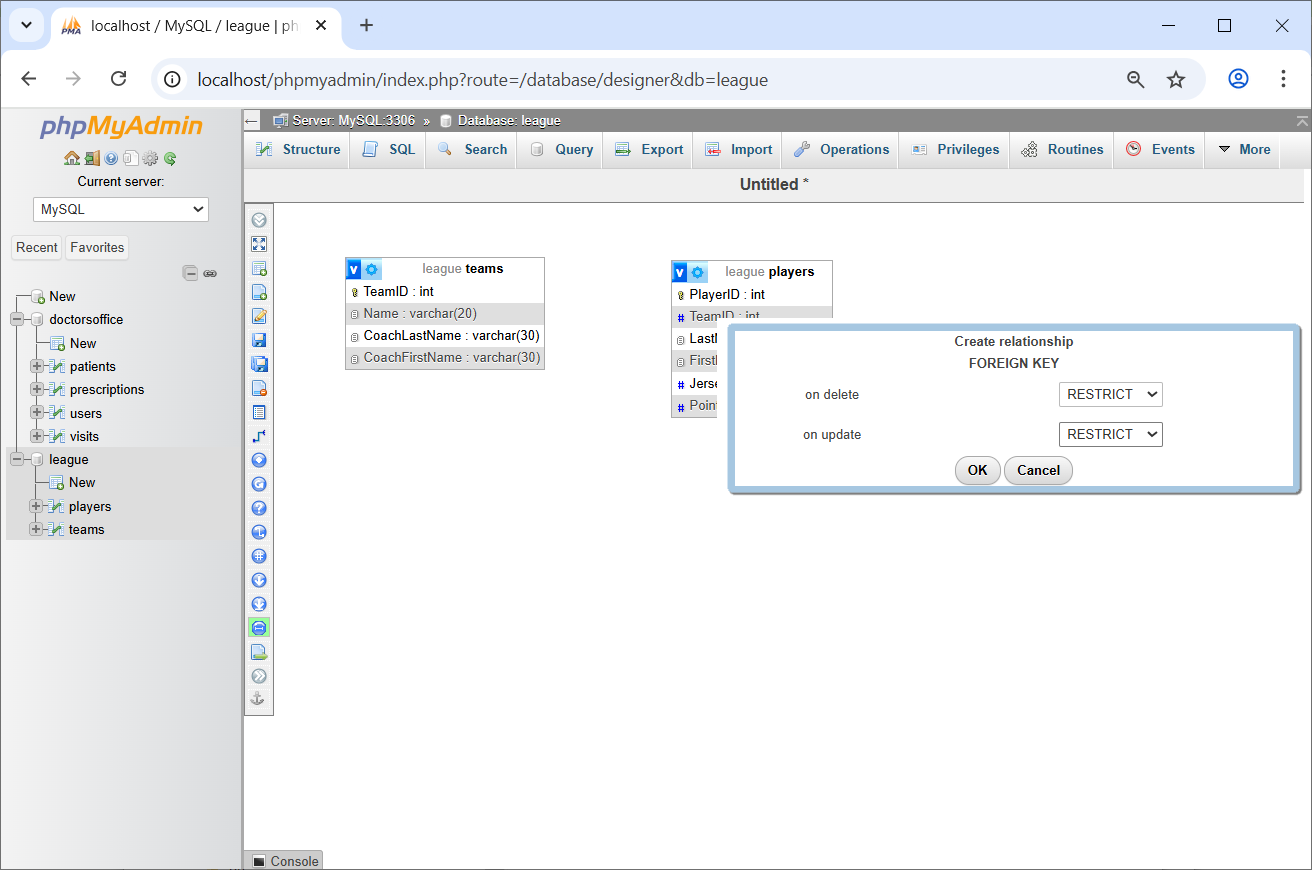
* 1. Select the *league* database and choose: More (far right top), Designer. The result is shown below.



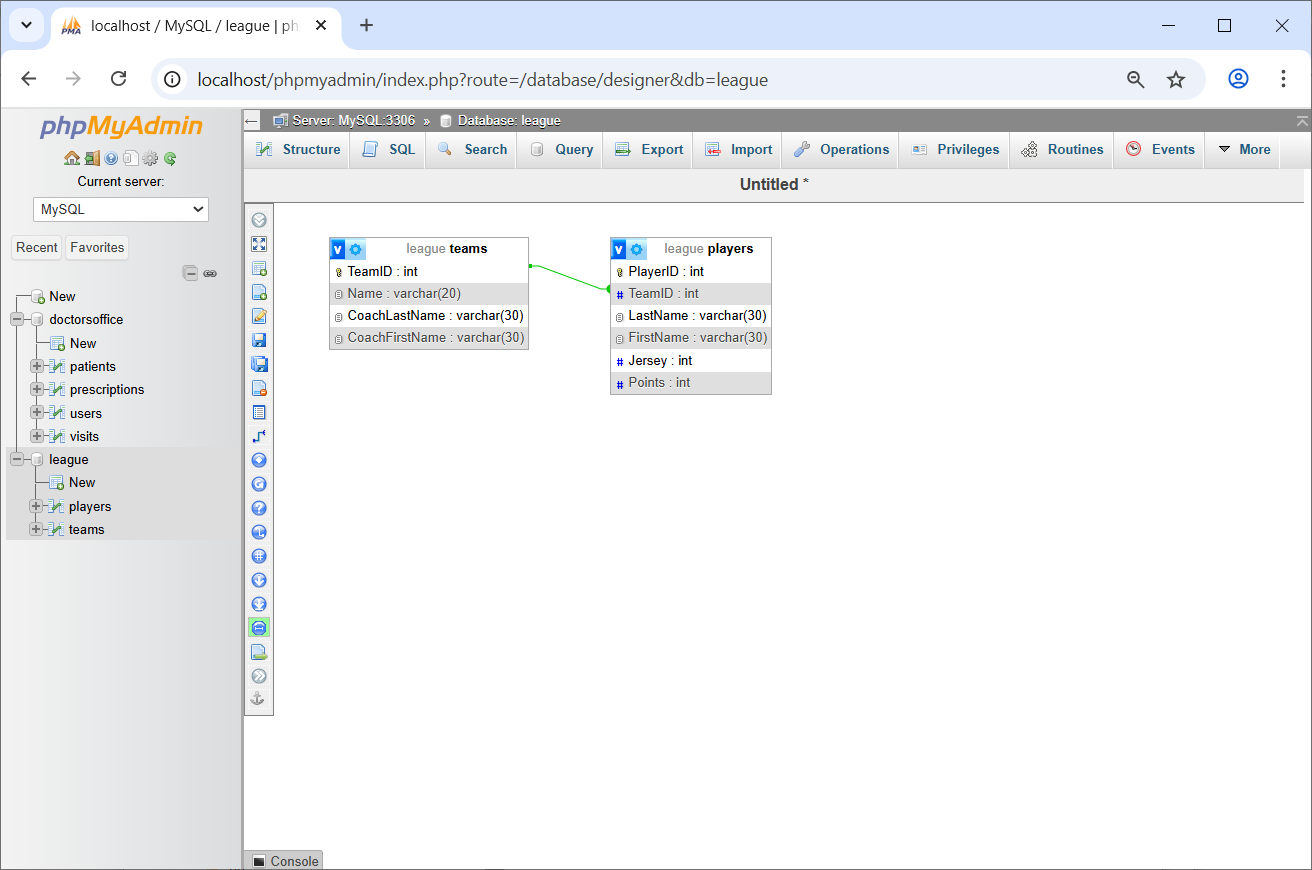
* 1. From the left menu, choose: Create relationship



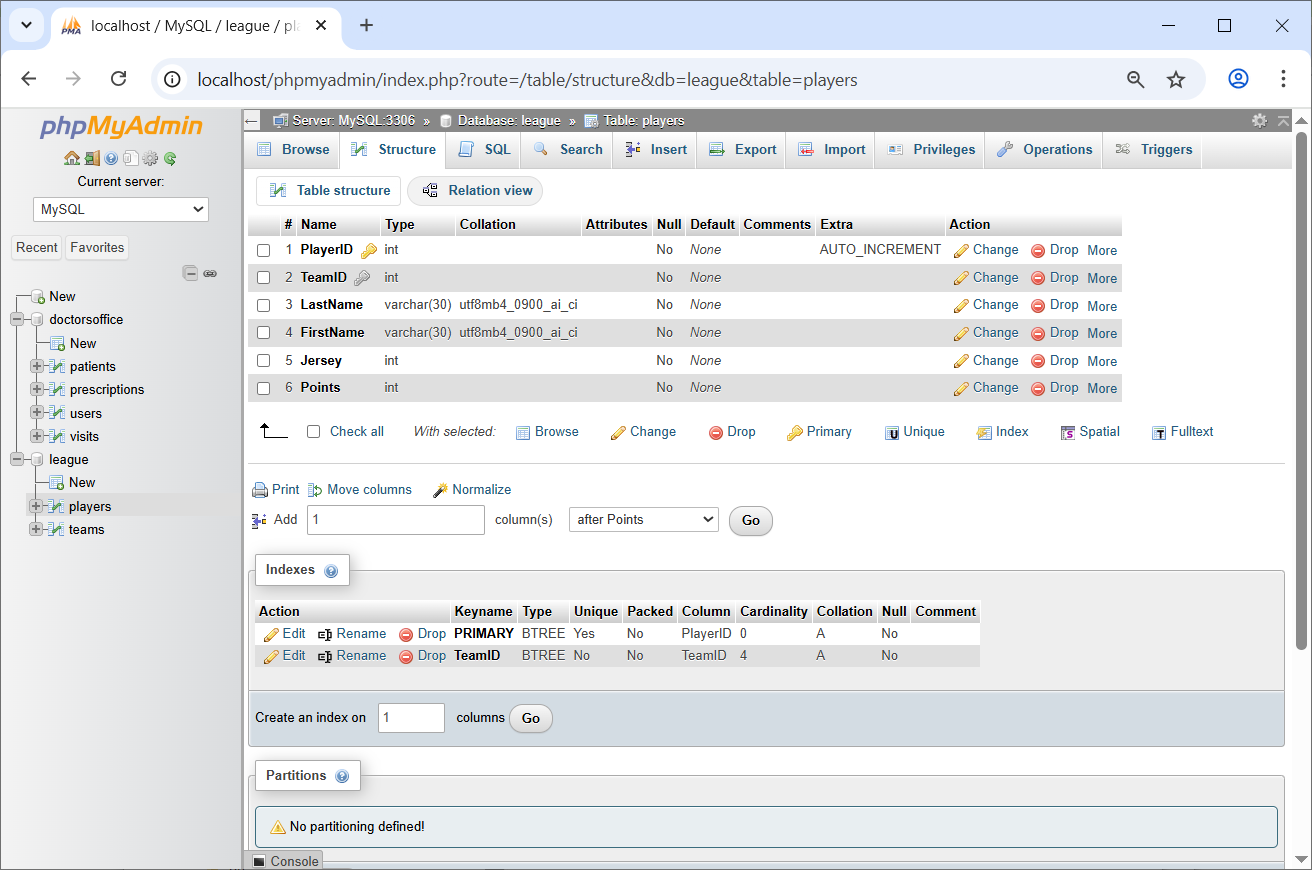
* 1. As the prompt leads you, click *TeamID* in the *teams* table, then click *TeamID* in the *players* table. On the pop-up, choose: Restrict for each of the options, then choose: OK.



* 1. The result will look like this. If the line showing the association is not showing. Select the database, then More, Designer to redisplay.



* 1. Select the *players* table and then choose: Structure. You will see that *TeamID* has a grey key beside it indicating that it is a foreign key.



***The End…***

1. <https://stackoverflow.com/questions/1108/how-does-database-indexing-work/1130#1130> [↑](#footnote-ref-1)