Muscat College

Database Principles and Applications

Code: CSCUMB3

Assignment 2020

Submitted by

Stirling No:_2840080

Submitted on

27-Nov-2020

Database Principles and Applications

Part B

Task 1: MySQL database implementation

- **Create table of the rowdata.csv:**

Table Team			
No	Columns	Datatype	Constraints
1	Team Name	Varchar(8)	Primary Key
2	Town	Varchar(15)	NOT NULL
Table Player			
No	Columns	Datatype	Constraints
1	Player ID	Int(5)	Primary key
2	Forename	Varchar(11)	NOT NULL
3	Surname	Varchar(12)	NOT NULL
4	Team	Varchar(8)	Foreign Key
5	Status	Varchar(12)	NOT NULL
Table Game			
No	Columns	Datatype	Constraints
1	Venue	Varchar(15)	NOT NULL
2	Date	Varchar(10)	Primary Key
Table teamGames			
No	Columns	Datatype	Constraints
1	TeamName	Varchar(8)	Primary Key, Foreign Key
2	Date	Varchar(10)	Primary Key, Foreign Key

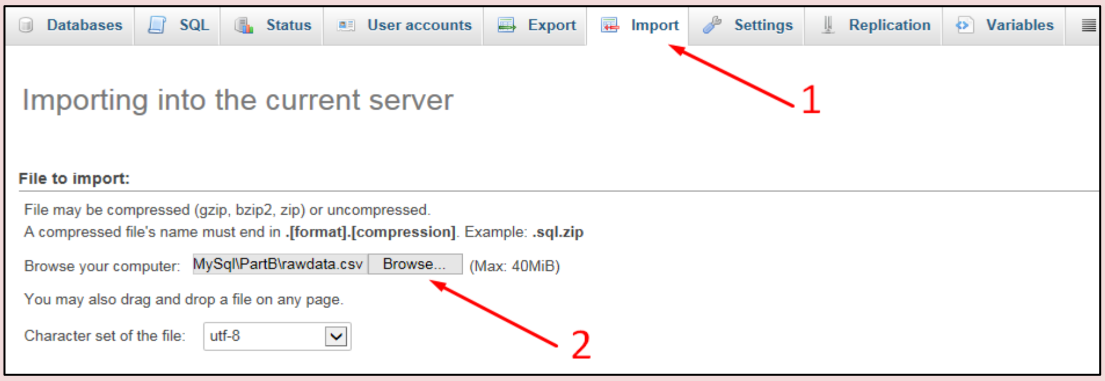
Table Skills			
No	Columns	Datatype	Constraints
1	Player ID	Int(5)	Primary Key, Foreign Key
2	Skill	Varchar(9)	Primary Key, Foreign Key

Table Points			
No	Columns	Datatype	Constraints
1	Player ID	Int(5)	Primary Key, Foreign Key
2	Date	Varchar(10)	Primary Key, Foreign Key
3	Points	Int(1)	NOT NULL

1) Import rowdata.csv:

To import rowdata.csv into the database:

- go to “Import” option.
- browse for the csv file.
- Select format “CSV” from dropdown list
- Check the option: “The first line of the file contains the table column names”



Importing into the current server

File to import:

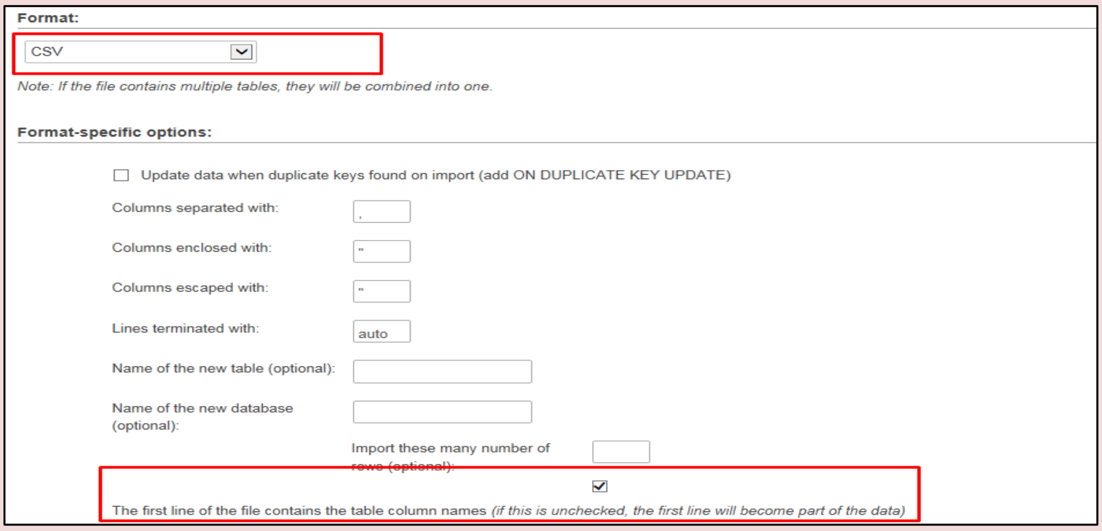
File may be compressed (gzip, bzip2, zip) or uncompressed.
A compressed file's name must end in **.[format].[compression]**. Example: **.sql.zip**

Browse your computer: MySql\PartB\rawdata.csv Browse... (Max: 40MiB)

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

Character set of the file: utf-8

Figure 1-Import rowdata.csv



Format:

CSV

Note: If the file contains multiple tables, they will be combined into one.

Format-specific options:

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

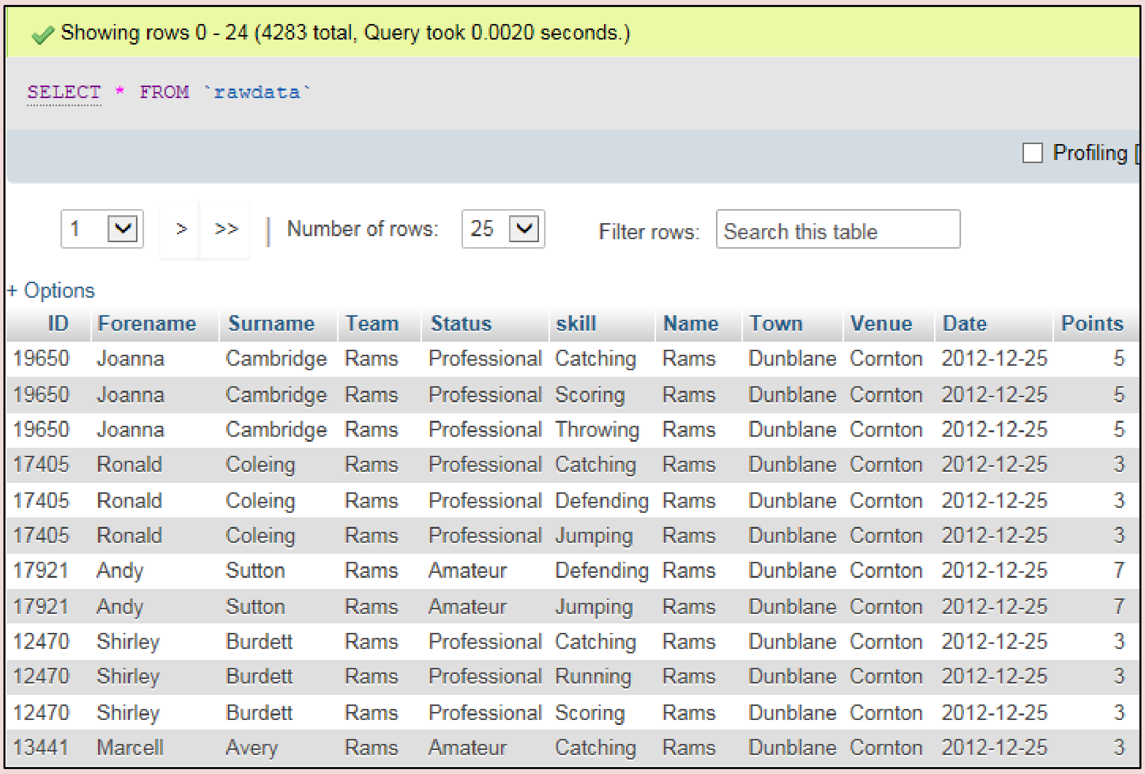
Columns separated with:	,
Columns enclosed with:	"
Columns escaped with:	"
Lines terminated with:	auto
Name of the new table (optional):	
Name of the new database (optional):	
Import these many number of rows (optional):	

☑ The first line of the file contains the table column names *(if this is unchecked, the first line will become part of the data)*

Figure 2- Select format “CSV”

When click on “Go” button, here the screen result showing that import was successful. To check if data is loaded, here the screen showing data:

```
SELECT * FROM 'rawdata'
```


Showing rows 0 - 24 (4283 total, Query took 0.0020 seconds.)

```
SELECT * FROM `rawdata`
```

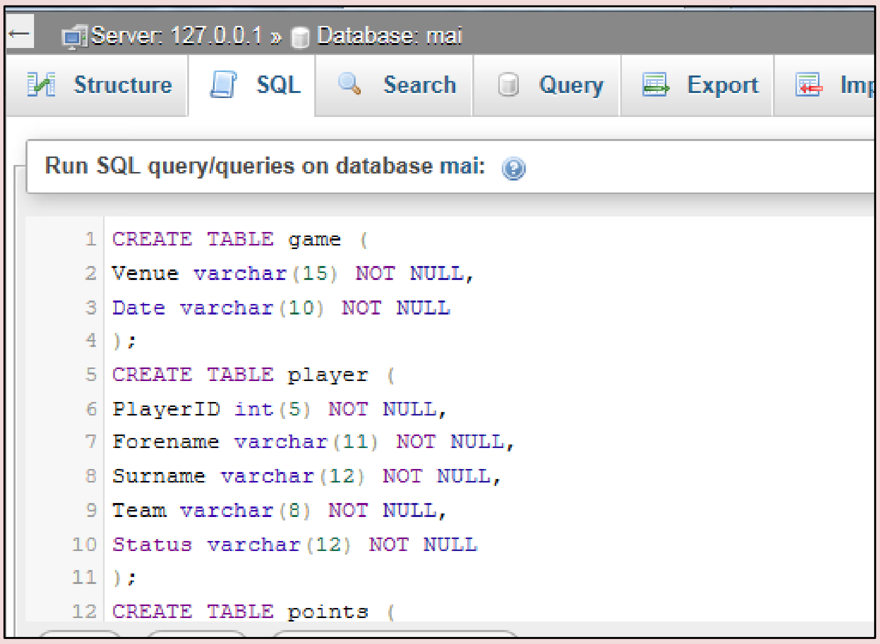
ID	Forename	Surname	Team	Status	skill	Name	Town	Venue	Date	Points
19650	Joanna	Cambridge	Rams	Professional	Catching	Rams	Dunblane	Cornton	2012-12-25	5
19650	Joanna	Cambridge	Rams	Professional	Scoring	Rams	Dunblane	Cornton	2012-12-25	5
19650	Joanna	Cambridge	Rams	Professional	Throwing	Rams	Dunblane	Cornton	2012-12-25	5
17405	Ronald	Coleing	Rams	Professional	Catching	Rams	Dunblane	Cornton	2012-12-25	3
17405	Ronald	Coleing	Rams	Professional	Defending	Rams	Dunblane	Cornton	2012-12-25	3
17405	Ronald	Coleing	Rams	Professional	Jumping	Rams	Dunblane	Cornton	2012-12-25	3
17921	Andy	Sutton	Rams	Amateur	Defending	Rams	Dunblane	Cornton	2012-12-25	7
17921	Andy	Sutton	Rams	Amateur	Jumping	Rams	Dunblane	Cornton	2012-12-25	7
12470	Shirley	Burdett	Rams	Professional	Catching	Rams	Dunblane	Cornton	2012-12-25	3
12470	Shirley	Burdett	Rams	Professional	Running	Rams	Dunblane	Cornton	2012-12-25	3
12470	Shirley	Burdett	Rams	Professional	Scoring	Rams	Dunblane	Cornton	2012-12-25	3
13441	Marcell	Avery	Rams	Amateur	Catching	Rams	Dunblane	Cornton	2012-12-25	3

Figure 3- import was successful

2) SQL Code to create 3NF Tables:

```
CREATE TABLE game (
Venue varchar(15) NOT NULL,
Date varchar(10) NOT NULL
);
CREATE TABLE player (
PlayerID int(5) NOT NULL,
Forename varchar(11) NOT NULL,
Surname varchar(12) NOT NULL,
Team varchar(8) NOT NULL,
Status varchar(12) NOT NULL
);
CREATE TABLE points (
PlayerID int(5) NOT NULL,
Date varchar(10) NOT NULL,
Points int(1) NOT NULL
);
CREATE TABLE skills (
PlayerID int(5) NOT NULL,
Skill varchar(9) NOT NULL
);
CREATE TABLE team (
TeamName varchar(8) NOT NULL,
Town varchar(15) NOT NULL
);
CREATE TABLE teamgames (
TeamName varchar(8) NOT NULL,
Date varchar(10) NOT NULL
);
ALTER TABLE game
ADD PRIMARY KEY (Date);

ALTER TABLE player
ADD PRIMARY KEY (PlayerID);

ALTER TABLE points
ADD PRIMARY KEY (PlayerID,Date);
```

```
ALTER TABLE skills

ADD PRIMARY KEY (PlayerID,Skill);


ALTER TABLE team

ADD PRIMARY KEY (TeamName);


ALTER TABLE teamgames

ADD PRIMARY KEY (TeamName,Date);
```


Server: 127.0.0.1 » Database: mai
Structure
SQL
Search
Query
Export
Imp
Run SQL query/queries on database mai:

```
CREATE TABLE game (
Venue varchar(15) NOT NULL,
Date varchar(10) NOT NULL
);
CREATE TABLE player (
PlayerID int(5) NOT NULL,
Forename varchar(11) NOT NULL,
Surname varchar(12) NOT NULL,
Team varchar(8) NOT NULL,
Status varchar(12) NOT NULL
);
CREATE TABLE points (
```

Figure 4-Inserting the codes into SQL.

This is the screen to show the created tables:

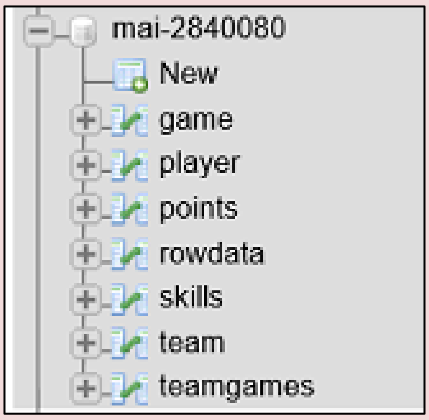
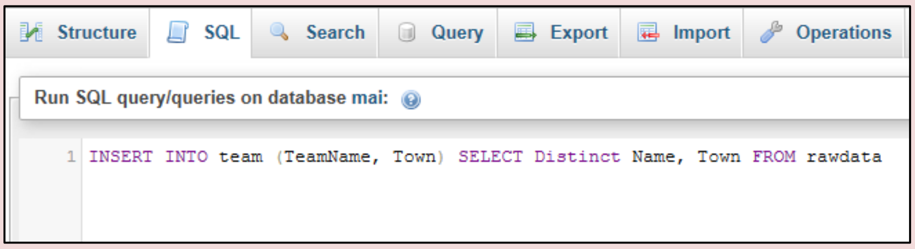


Figure 5-Tables inserted.

3) Loading data:

Load data to table "Team":

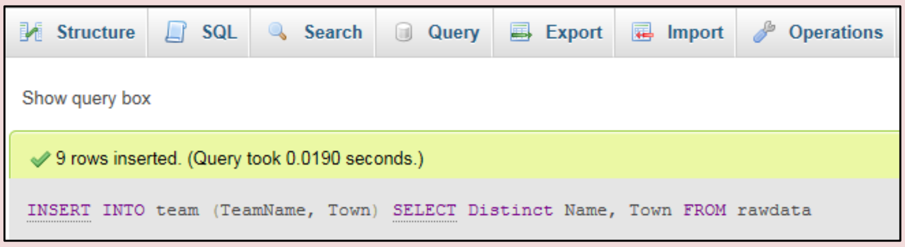
INSERT INTO team (TeamName, Town) SELECT Distinct Name, Town FROM rawdata;



Run SQL query/queries on database mai:

```
INSERT INTO team (TeamName, Town) SELECT Distinct Name, Town FROM rawdata
```

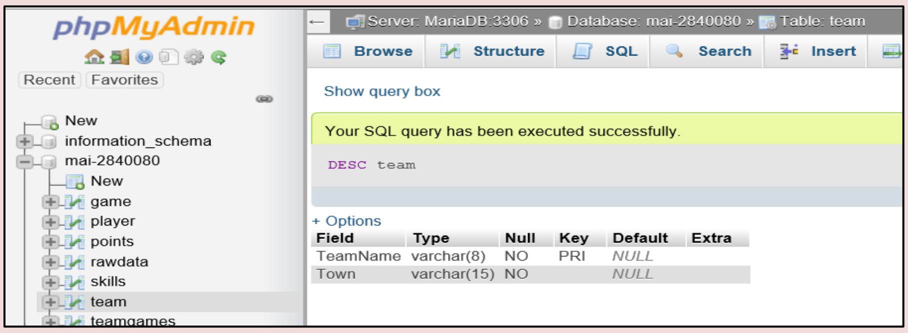
Figure 6-Load data for Team table.



✓ 9 rows inserted. (Query took 0.0190 seconds.)

```
INSERT INTO team (TeamName, Town) SELECT Distinct Name, Town FROM rawdata
```

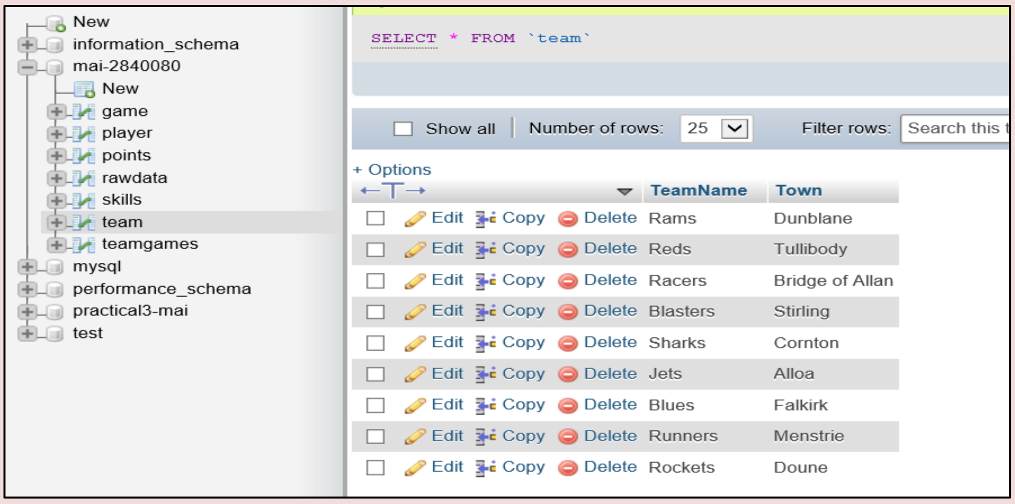
Figure 7-The data for Team table is inserted.



Your SQL query has been executed successfully.

```
DESC team
```

Field	Type	Null	Key	Default	Extra
TeamName	varchar(8)	NO	PRI	NULL	
Town	varchar(15)	NO		NULL	

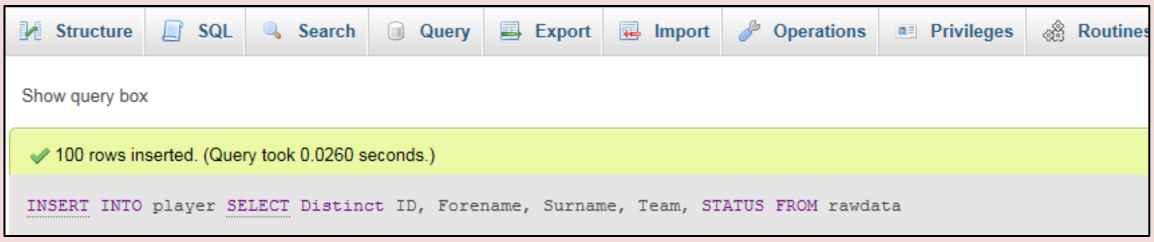


```
SELECT * FROM `team`
```

TeamName	Town
Rams	Dunblane
Reds	Tullibody
Racers	Bridge of Allan
Blasters	Stirling
Sharks	Cornton
Jets	Alloa
Blues	Falkirk
Runners	Menstrie
Rockets	Doune

Load data to table "Player":

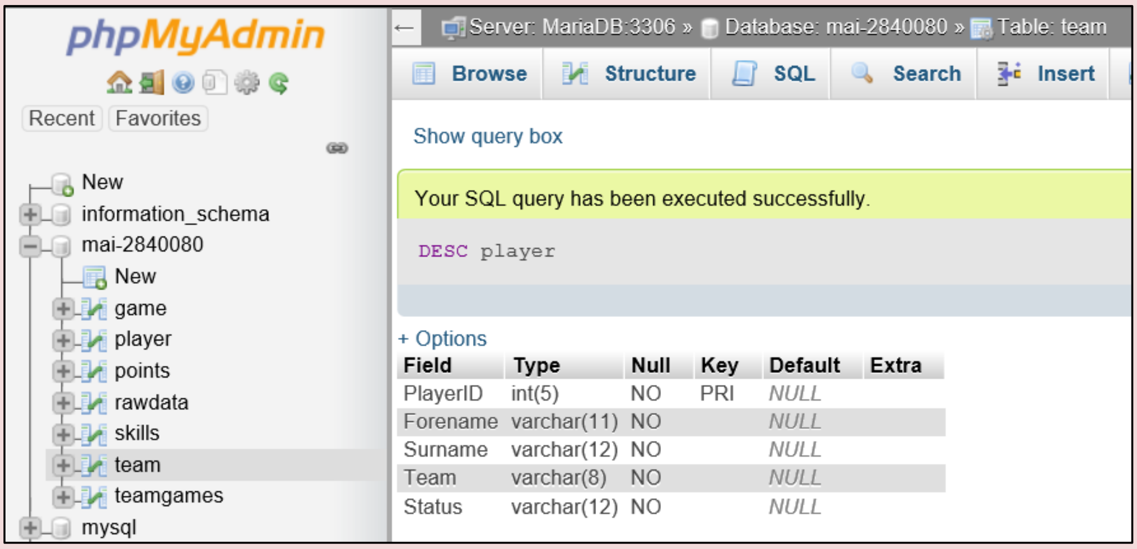
INSERT INTO player SELECT Distinct ID, Forename, Surname, Team, STATUS FROM rawdata;



✓ 100 rows inserted. (Query took 0.0260 seconds.)

```
INSERT INTO player SELECT Distinct ID, Forename, Surname, Team, STATUS FROM rawdata
```

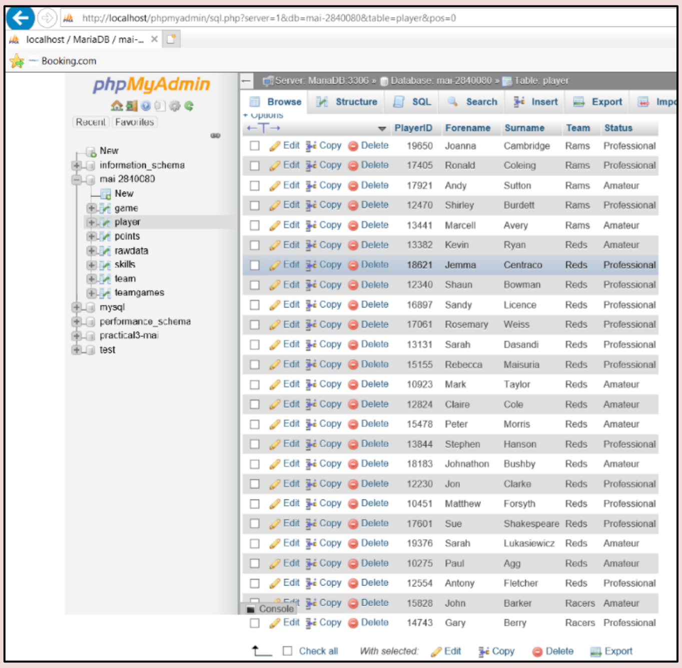
Figure 8- The data for Player table is inserted.



Your SQL query has been executed successfully.

```
DESC player
```

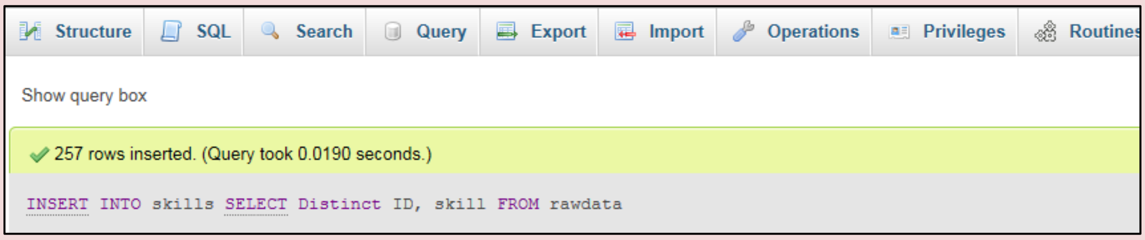
Field	Type	Null	Key	Default	Extra
PlayerID	int(5)	NO	PRI	NULL	
Forename	varchar(11)	NO		NULL	
Surname	varchar(12)	NO		NULL	
Team	varchar(8)	NO		NULL	
Status	varchar(12)	NO		NULL	



	PlayerID	Forename	Surname	Team	Status
Edit Copy Delete	19650	Joanna	Cambridge	Rams	Professional
Edit Copy Delete	17405	Ronald	Coleing	Rams	Professional
Edit Copy Delete	17921	Andy	Sutton	Rams	Amateur
Edit Copy Delete	12470	Shirley	Burdett	Rams	Professional
Edit Copy Delete	13441	Marcell	Avery	Rams	Amateur
Edit Copy Delete	13382	Kevin	Ryan	Reds	Amateur
Edit Copy Delete	18621	Jemma	Centraco	Reds	Professional
Edit Copy Delete	12340	Shaun	Bowman	Reds	Professional
Edit Copy Delete	16897	Sandy	Licence	Reds	Professional
Edit Copy Delete	17061	Rosemary	Weiss	Reds	Professional
Edit Copy Delete	13131	Sarah	Dasandi	Reds	Professional
Edit Copy Delete	15155	Rebecca	Maisuria	Reds	Professional
Edit Copy Delete	10923	Mark	Taylor	Reds	Amateur
Edit Copy Delete	12824	Claire	Cole	Reds	Amateur
Edit Copy Delete	15478	Peter	Morris	Reds	Amateur
Edit Copy Delete	13844	Stephen	Hanson	Reds	Professional
Edit Copy Delete	18183	Johnathon	Bushby	Reds	Amateur
Edit Copy Delete	12230	Jon	Clarke	Reds	Professional
Edit Copy Delete	10451	Matthew	Forsyth	Reds	Professional
Edit Copy Delete	17601	Sue	Shakespeare	Reds	Professional
Edit Copy Delete	19376	Sarah	Lukasiewicz	Reds	Amateur
Edit Copy Delete	10275	Paul	Agg	Reds	Amateur
Edit Copy Delete	12554	Antony	Fletcher	Reds	Professional
Edit Copy Delete	15828	John	Barker	Racers	Amateur
Edit Copy Delete	14743	Gary	Berry	Racers	Professional

Load data to table “Skills”:

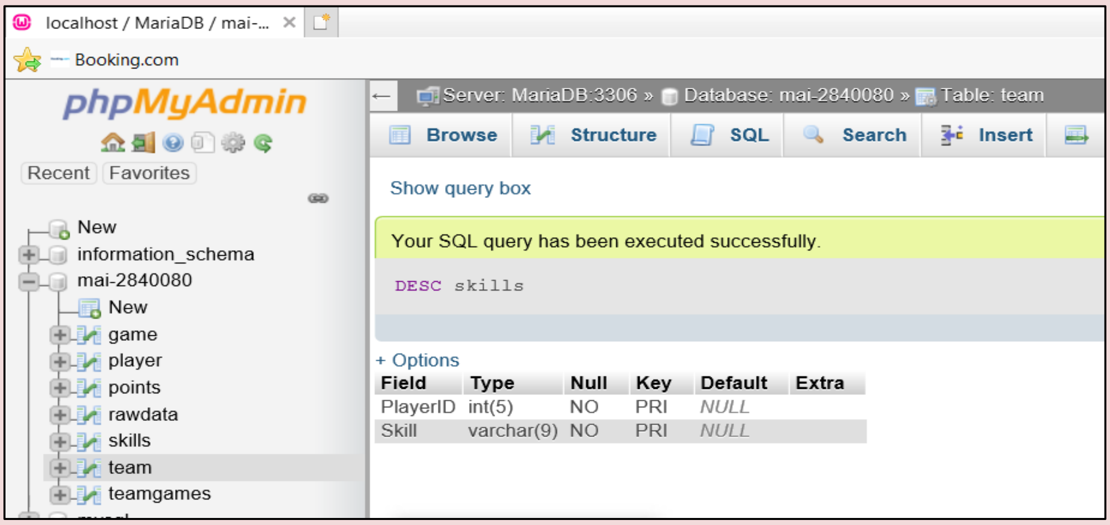
INSERT INTO skills SELECT Distinct ID, skill FROM rawdata;



✓ 257 rows inserted. (Query took 0.0190 seconds.)

```
INSERT INTO skills SELECT Distinct ID, skill FROM rawdata
```

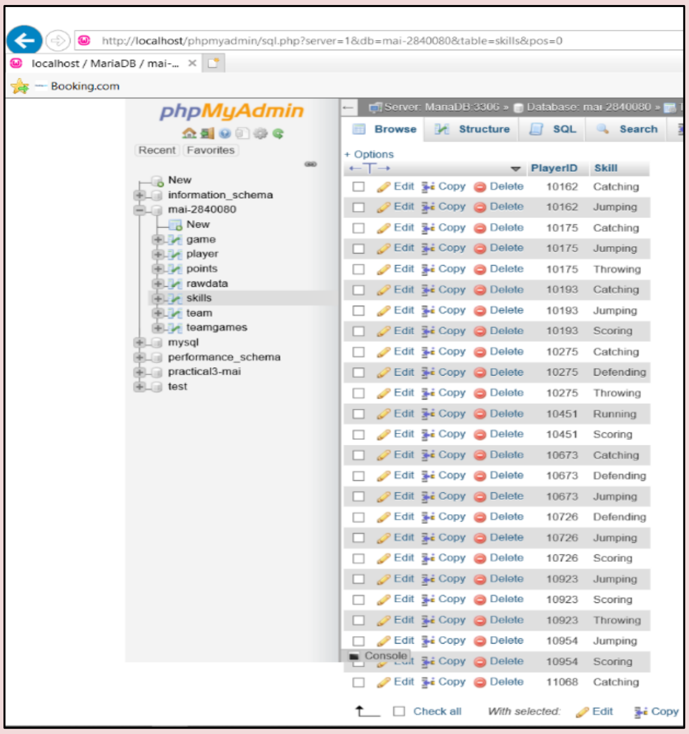
Figure 9- The data for Skills table is inserted.



Your SQL query has been executed successfully.

```
DESC skills
```

Field	Type	Null	Key	Default	Extra
PlayerID	int(5)	NO	PRI	*NULL*	
Skill	varchar(9)	NO	PRI	*NULL*	

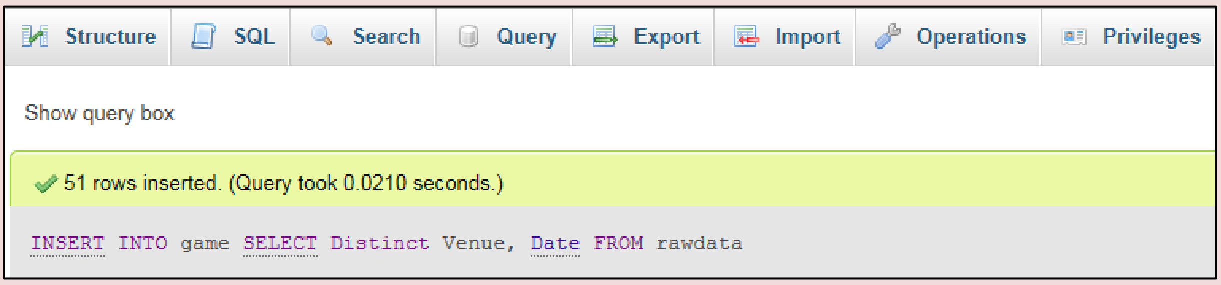


	PlayerID	Skill
Edit Copy Delete	10162	Catching
Edit Copy Delete	10162	Jumping
Edit Copy Delete	10175	Catching
Edit Copy Delete	10175	Jumping
Edit Copy Delete	10175	Throwing
Edit Copy Delete	10193	Catching
Edit Copy Delete	10193	Jumping
Edit Copy Delete	10193	Scoring
Edit Copy Delete	10275	Catching
Edit Copy Delete	10275	Defending
Edit Copy Delete	10275	Throwing
Edit Copy Delete	10451	Running
Edit Copy Delete	10451	Scoring
Edit Copy Delete	10673	Catching
Edit Copy Delete	10673	Defending
Edit Copy Delete	10673	Jumping
Edit Copy Delete	10726	Defending
Edit Copy Delete	10726	Jumping
Edit Copy Delete	10726	Scoring
Edit Copy Delete	10923	Jumping
Edit Copy Delete	10923	Scoring
Edit Copy Delete	10923	Throwing
Edit Copy Delete	10954	Jumping
Edit Copy Delete	10954	Scoring
Edit Copy Delete	11068	Catching

Check all *With selected:* Edit Copy

Load data to table "Game":

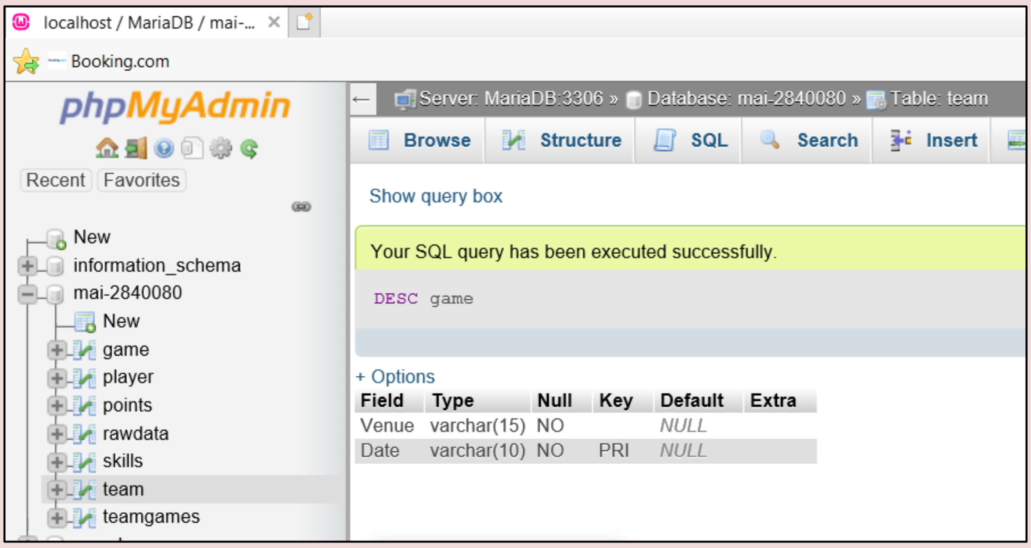
INSERT INTO game SELECT Distinct Venue, Date FROM rawdata;



✓ 51 rows inserted. (Query took 0.0210 seconds.)

```
INSERT INTO game SELECT Distinct Venue, Date FROM rawdata
```

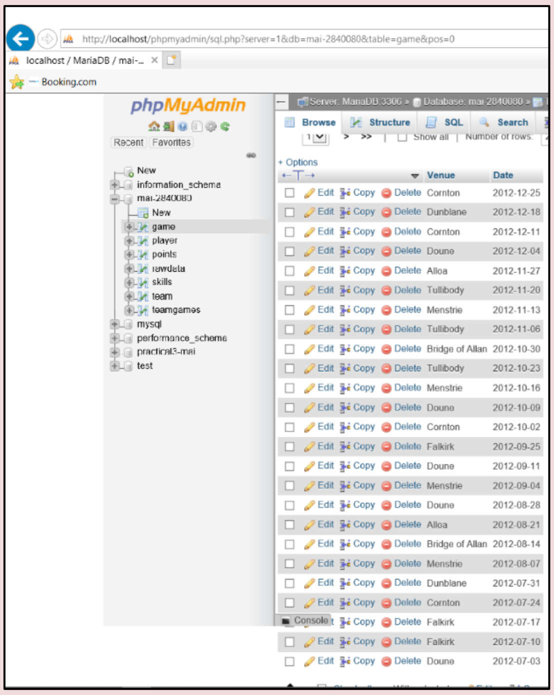
Figure 10-The data for Game table is inserted.



Your SQL query has been executed successfully.

```
DESC game
```

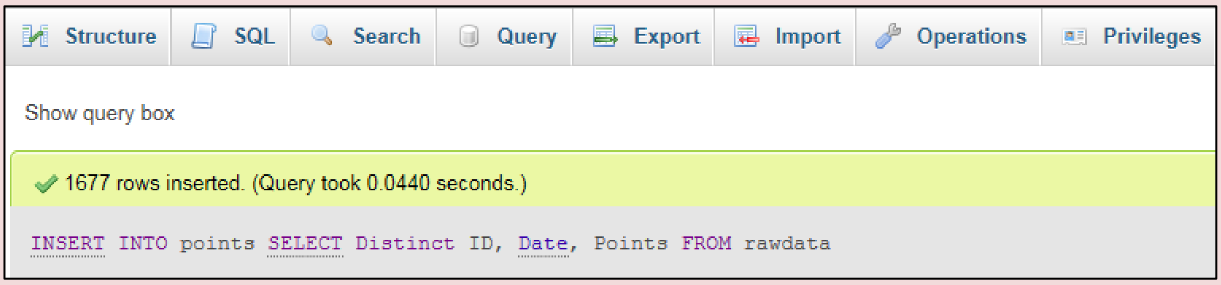
Field	Type	Null	Key	Default	Extra
Venue	varchar(15)	NO		NULL	
Date	varchar(10)	NO	PRI	NULL	



	Venue	Date
Edit Copy Delete	Cornton	2012-12-25
Edit Copy Delete	Dunblane	2012-12-18
Edit Copy Delete	Cornton	2012-12-11
Edit Copy Delete	Doune	2012-12-04
Edit Copy Delete	Alloa	2012-11-27
Edit Copy Delete	Tullibody	2012-11-20
Edit Copy Delete	Menstrie	2012-11-13
Edit Copy Delete	Tullibody	2012-11-06
Edit Copy Delete	Bridge of Allan	2012-10-30
Edit Copy Delete	Tullibody	2012-10-23
Edit Copy Delete	Menstrie	2012-10-16
Edit Copy Delete	Doune	2012-10-09
Edit Copy Delete	Cornton	2012-10-02
Edit Copy Delete	Falkirk	2012-09-25
Edit Copy Delete	Doune	2012-09-11
Edit Copy Delete	Menstrie	2012-09-04
Edit Copy Delete	Doune	2012-08-28
Edit Copy Delete	Alloa	2012-08-21
Edit Copy Delete	Bridge of Allan	2012-08-14
Edit Copy Delete	Menstrie	2012-08-07
Edit Copy Delete	Dunblane	2012-07-31
Edit Copy Delete	Cornton	2012-07-24
Copy Delete	Falkirk	2012-07-17
Edit Copy Delete	Falkirk	2012-07-10
Edit Copy Delete	Doune	2012-07-03

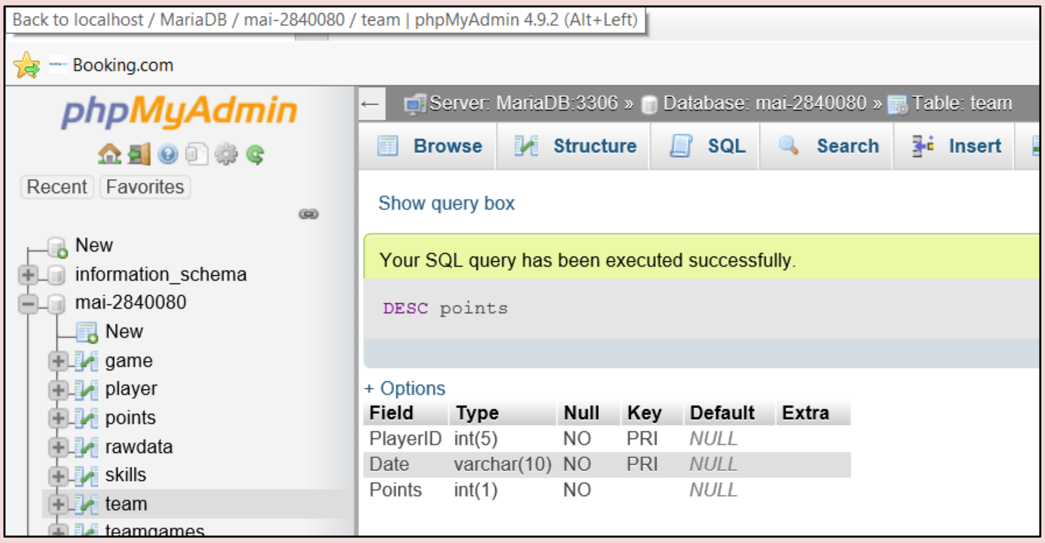
Load data to table "Points":

INSERT INTO points SELECT Distinct ID, Date, Points FROM rawdata;



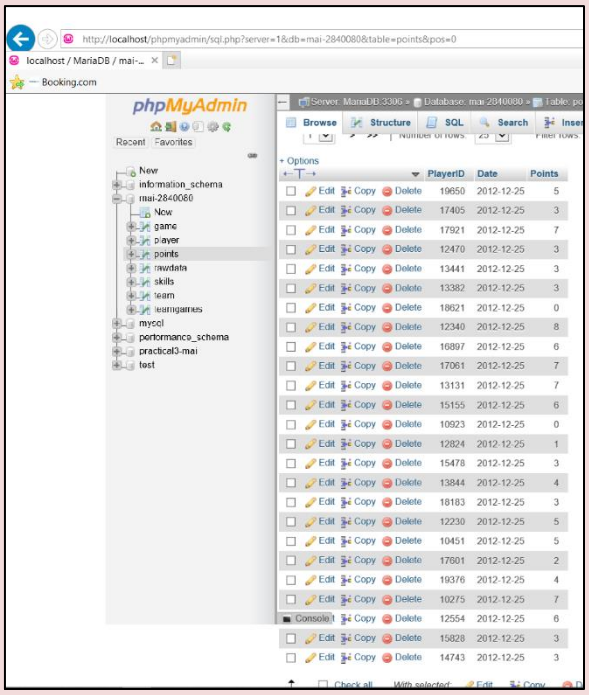
```
1677 rows inserted. (Query took 0.0440 seconds.)
INSERT INTO points SELECT Distinct ID, Date, Points FROM rawdata
```

Figure 11-The data for Points table is inserted.



```
Your SQL query has been executed successfully.
DESC points
```

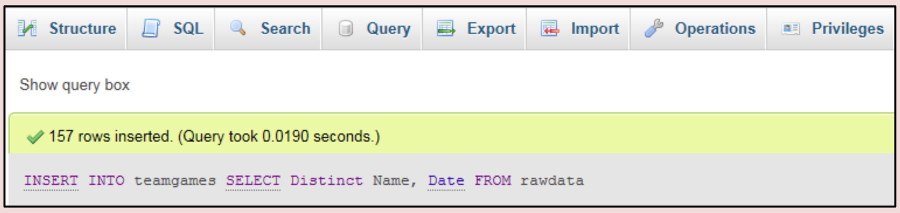
Field	Type	Null	Key	Default	Extra
PlayerID	int(5)	NO	PRI	NULL	
Date	varchar(10)	NO	PRI	NULL	
Points	int(1)	NO		NULL	



	PlayerID	Date	Points
Edit Copy Delete	19650	2012-12-25	5
Edit Copy Delete	17405	2012-12-25	3
Edit Copy Delete	17921	2012-12-25	7
Edit Copy Delete	12470	2012-12-25	3
Edit Copy Delete	13441	2012-12-25	3
Edit Copy Delete	13382	2012-12-25	3
Edit Copy Delete	18621	2012-12-25	0
Edit Copy Delete	12340	2012-12-25	8
Edit Copy Delete	16897	2012-12-25	6
Edit Copy Delete	17061	2012-12-25	7
Edit Copy Delete	13131	2012-12-25	7
Edit Copy Delete	15155	2012-12-25	6
Edit Copy Delete	10923	2012-12-25	0
Edit Copy Delete	12824	2012-12-25	1
Edit Copy Delete	15478	2012-12-25	3
Edit Copy Delete	13844	2012-12-25	4
Edit Copy Delete	18183	2012-12-25	3
Edit Copy Delete	12230	2012-12-25	5
Edit Copy Delete	10451	2012-12-25	5
Edit Copy Delete	17601	2012-12-25	2
Edit Copy Delete	19376	2012-12-25	4
Edit Copy Delete	10275	2012-12-25	7
t Copy Delete	12554	2012-12-25	6
Edit Copy Delete	15828	2012-12-25	3
Edit Copy Delete	14743	2012-12-25	3

Load data to table “TeamGames”:

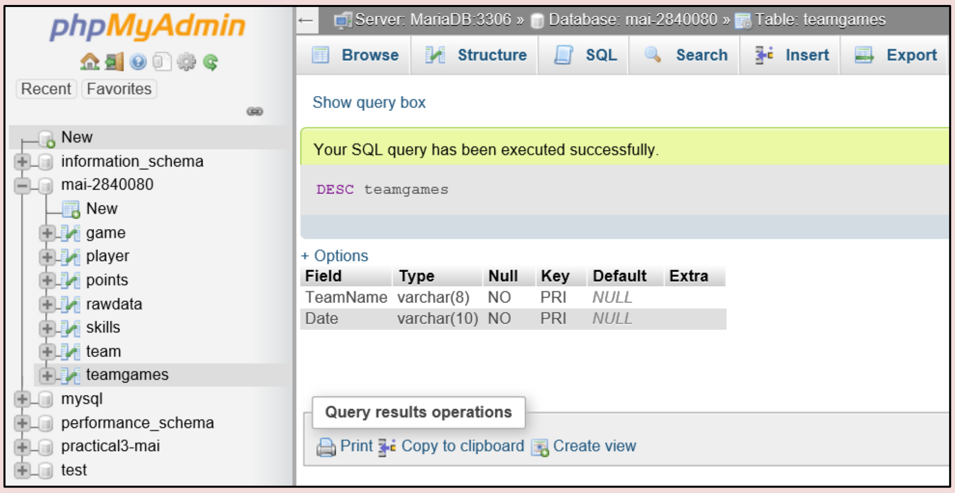
INSERT INTO teamgames SELECT Distinct Name, Date FROM rawdata;



✓ 157 rows inserted. (Query took 0.0190 seconds.)

```
INSERT INTO teamgames SELECT Distinct Name, Date FROM rawdata
```

Figure 12-The data for TeamGame table is inserted.

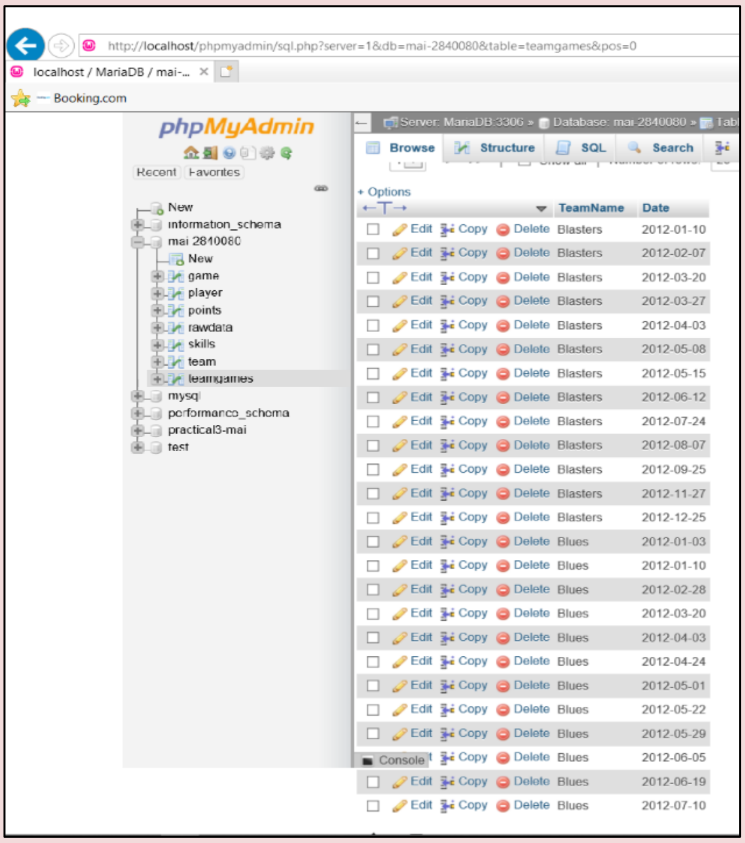


Your SQL query has been executed successfully.

```
DESC teamgames
```

+ Options

Field	Type	Null	Key	Default	Extra
TeamName	varchar(8)	NO	PRI	NULL	
Date	varchar(10)	NO	PRI	NULL	

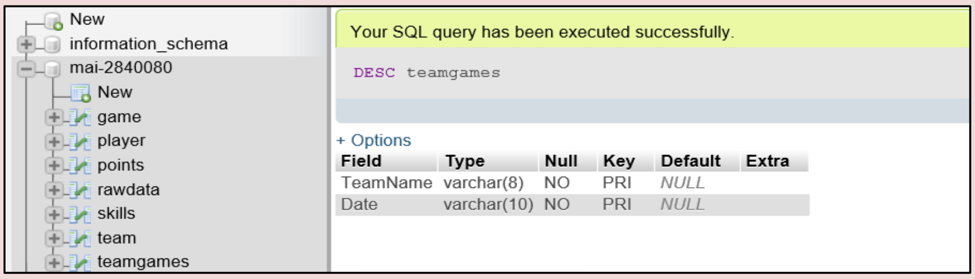


+ Options

	TeamName	Date
Edit Copy Delete	Blasters	2012-01-10
Edit Copy Delete	Blasters	2012-02-07
Edit Copy Delete	Blasters	2012-03-20
Edit Copy Delete	Blasters	2012-03-27
Edit Copy Delete	Blasters	2012-04-03
Edit Copy Delete	Blasters	2012-05-08
Edit Copy Delete	Blasters	2012-05-15
Edit Copy Delete	Blasters	2012-06-12
Edit Copy Delete	Blasters	2012-07-24
Edit Copy Delete	Blasters	2012-08-07
Edit Copy Delete	Blasters	2012-09-25
Edit Copy Delete	Blasters	2012-11-27
Edit Copy Delete	Blasters	2012-12-25
Edit Copy Delete	Blues	2012-01-03
Edit Copy Delete	Blues	2012-01-10
Edit Copy Delete	Blues	2012-02-28
Edit Copy Delete	Blues	2012-03-20
Edit Copy Delete	Blues	2012-04-03
Edit Copy Delete	Blues	2012-04-24
Edit Copy Delete	Blues	2012-05-01
Edit Copy Delete	Blues	2012-05-22
Edit Copy Delete	Blues	2012-05-29
Copy Delete	Blues	2012-06-05
Edit Copy Delete	Blues	2012-06-19
Edit Copy Delete	Blues	2012-07-10

● REFERENCES Function

-
  ```
  CREATE TABLE teamgames;
  (TeamName Varchar(8) REFERENCES Team,
  Date VARCHAR (10) REFERENCES Game);
  DESC teamgames;
  ```

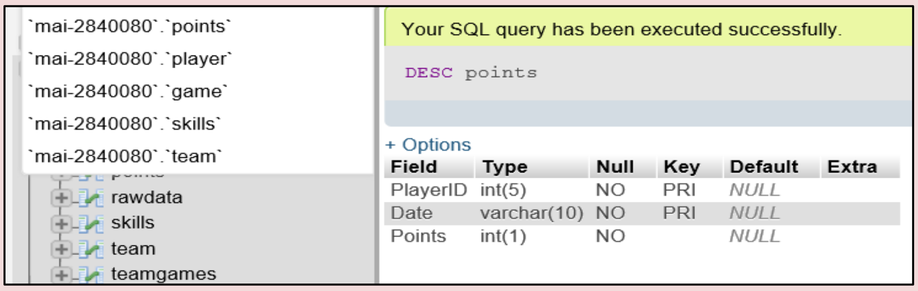


Your SQL query has been executed successfully.

```
DESC teamgames
```

Field	Type	Null	Key	Default	Extra
TeamName	varchar(8)	NO	PRI	NULL	
Date	varchar(10)	NO	PRI	NULL	

-
  ```
  CREATE TABLE points;
  (playerID int(5) REFERENCES player,
  Date VARCHAR (9) REFERENCES Game);
  DESC Points;
  ```

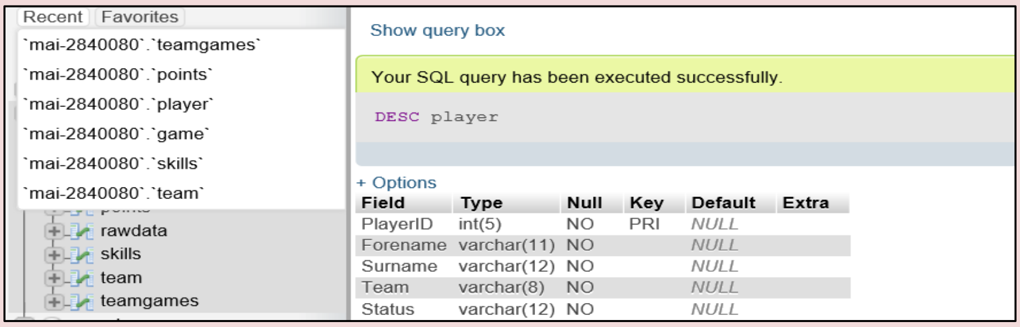


Your SQL query has been executed successfully.

```
DESC points
```

Field	Type	Null	Key	Default	Extra
PlayerID	int(5)	NO	PRI	NULL	
Date	varchar(10)	NO	PRI	NULL	
Points	int(1)	NO		NULL	

-
  ```
  CREATE TABLE player;
  (Forename VARCHAR(11),
  Surename VARCHAR(12),
  playerID int(5) REFERENCES skills,
  Team VARCHAR (8) REFERENCES player,
  Status VARCHAR(10)); DESC player;
  ```



Your SQL query has been executed successfully.

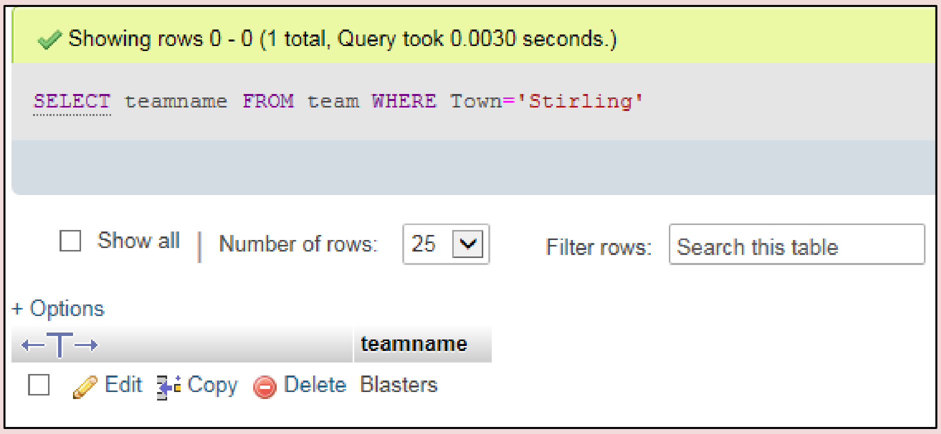
```
DESC player
```

Field	Type	Null	Key	Default	Extra
PlayerID	int(5)	NO	PRI	NULL	
Forename	varchar(11)	NO		NULL	
Surname	varchar(12)	NO		NULL	
Team	varchar(8)	NO		NULL	
Status	varchar(12)	NO		NULL	

Task 2: Querying database

1) List the name of the team that is based in Stirling:

```
SELECT teamname FROM  team  WHERE Town='Stirling';
```



Showing rows 0 - 0 (1 total, Query took 0.0030 seconds.)

```
SELECT teamname FROM team WHERE Town='Stirling'
```

Show all | Number of rows: 25 Filter rows: Search this table

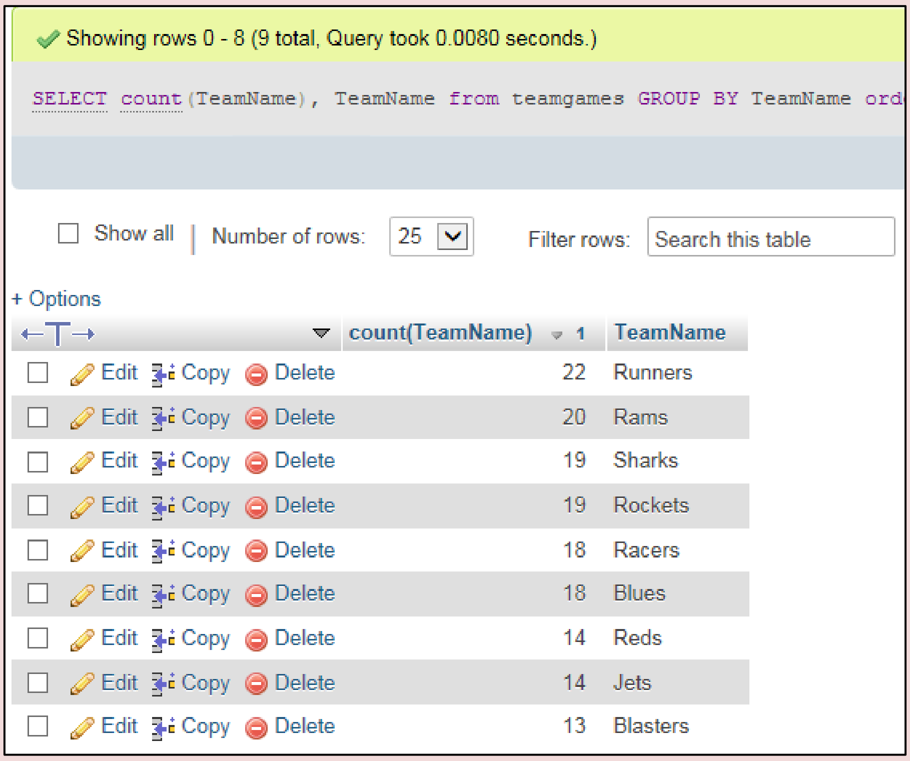
+ Options

	teamname
Edit Copy Delete	Blasters

Figure 13-Team name selected successfully.

2) List the total number of games played by each team, with the largest number first:

```
SELECT count(TeamName), TeamName from teamgames GROUP BY
TeamName order by count(TeamName) desc;
```



Showing rows 0 - 8 (9 total, Query took 0.0080 seconds.)

```
SELECT count(TeamName), TeamName from teamgames GROUP BY TeamName ord
```

Show all | Number of rows: 25 Filter rows: Search this table

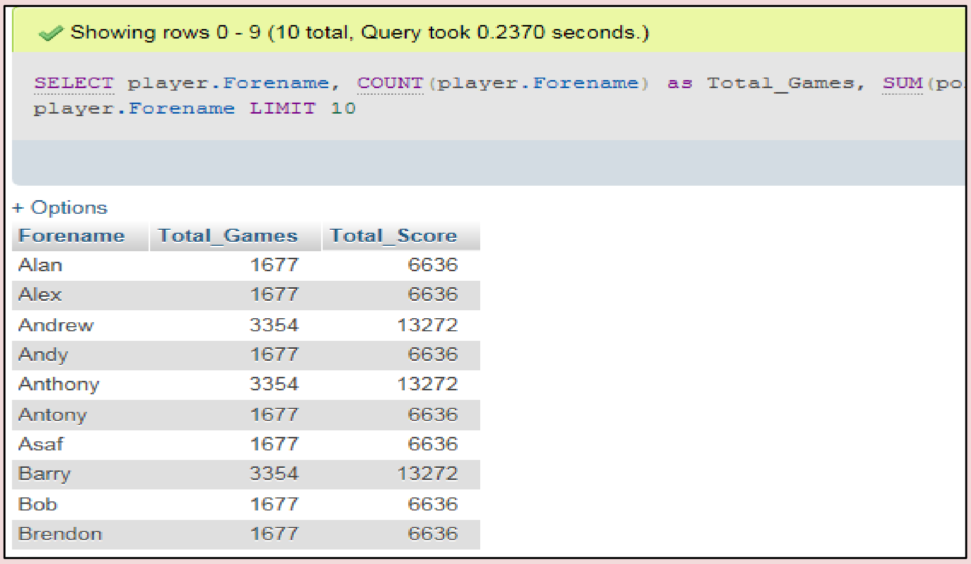
+ Options

	count(TeamName) 1	TeamName
Edit Copy Delete	22	Runners
Edit Copy Delete	20	Rams
Edit Copy Delete	19	Sharks
Edit Copy Delete	19	Rockets
Edit Copy Delete	18	Racers
Edit Copy Delete	18	Blues
Edit Copy Delete	14	Reds
Edit Copy Delete	14	Jets
Edit Copy Delete	13	Blasters

Figure 14-The total number and the largest number of matches were determined successfully.

3) List the total number of games played and the total points scored by each player (list player name plus total number of games and points scored, but just give the first 10 results in your report):

```
SELECT player.Forename, COUNT(player.Forename) as Total_Games,
SUM(points.Points) as Total_Score from player,points GROUP BY
player.Forename LIMIT 10;
```



✓ Showing rows 0 - 9 (10 total, Query took 0.2370 seconds.)

```
SELECT player.Forename, COUNT(player.Forename) as Total_Games, SUM(po
player.Forename LIMIT 10
```

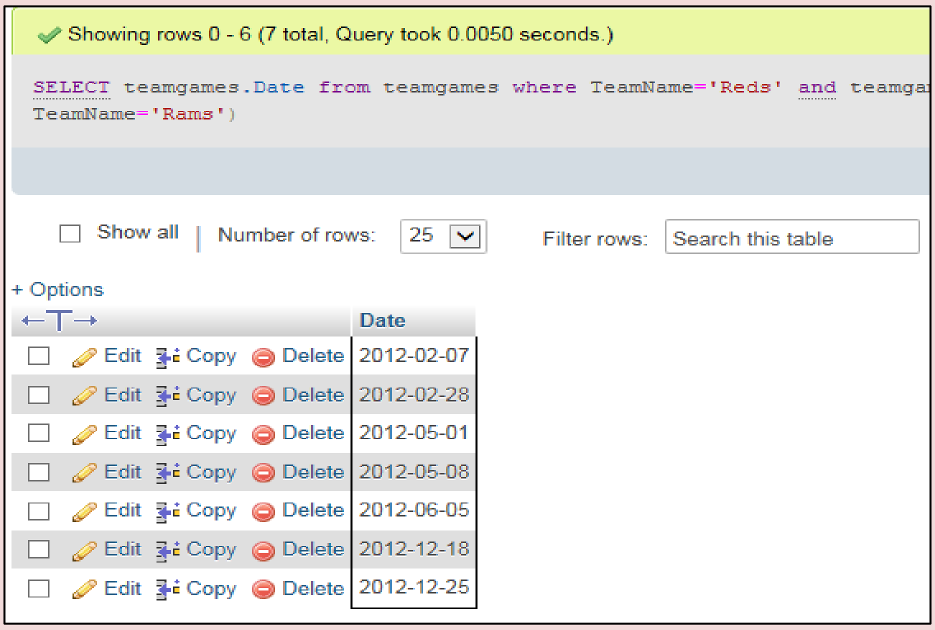
+ Options

Forename	Total_Games	Total_Score
Alan	1677	6636
Alex	1677	6636
Andrew	3354	13272
Andy	1677	6636
Anthony	3354	13272
Antony	1677	6636
Asaf	1677	6636
Barry	3354	13272
Bob	1677	6636
Brendon	1677	6636

Figure 15-The data results were shown successfully.

4) List the dates of all the games where the Reds and the Rams both played:

```
SELECT teamgames.Date from teamgames where TeamName='Reds' and
teamgames.Date in (SELECT teamgames.Date from teamgames where
TeamName='Rams');
```



✓ Showing rows 0 - 6 (7 total, Query took 0.0050 seconds.)

```
SELECT teamgames.Date from teamgames where TeamName='Reds' and teamga
TeamName='Rams')
```

Show all | Number of rows: 25 Filter rows: Search this table

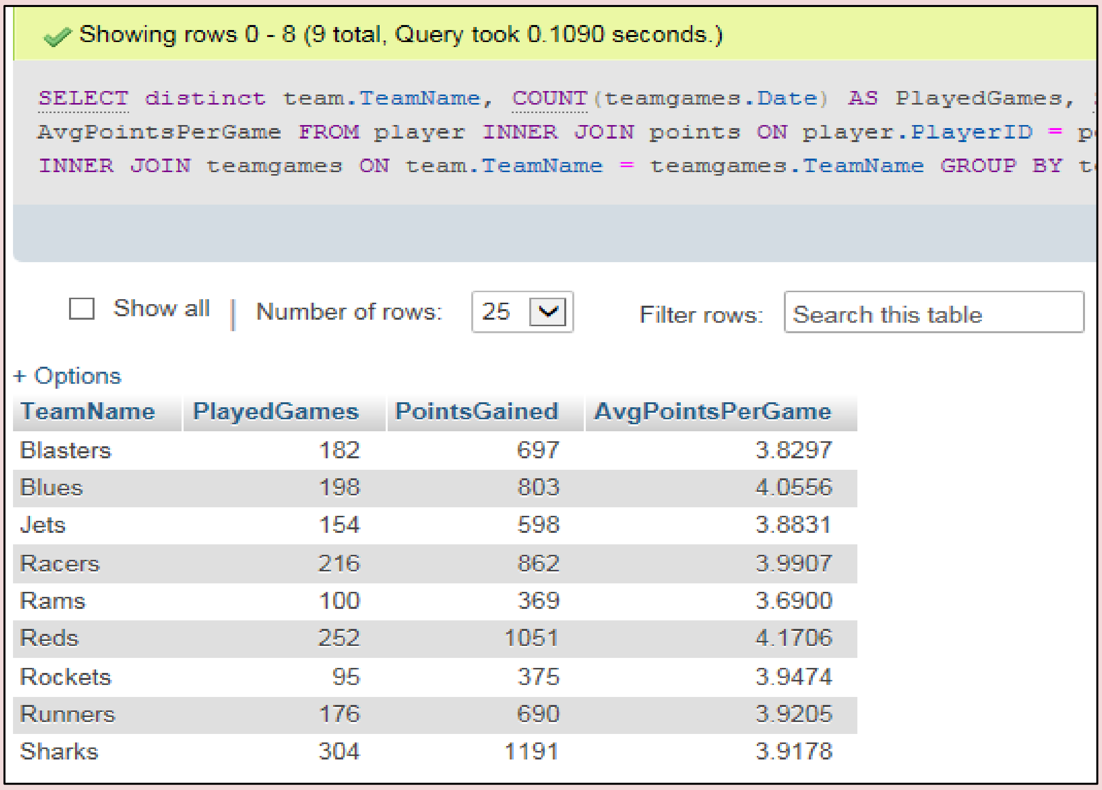
+ Options

	Date
Edit Copy Delete	2012-02-07
Edit Copy Delete	2012-02-28
Edit Copy Delete	2012-05-01
Edit Copy Delete	2012-05-08
Edit Copy Delete	2012-06-05
Edit Copy Delete	2012-12-18
Edit Copy Delete	2012-12-25

Figure 16-Show dates of all matches.

5) Produce the end of year team league table showing Team name, Number of games played, Number of points gained, Average points per game for each team:

```
SELECT distinct team.TeamName, COUNT(teamgames.Date) AS PlayedGames,
SUM(points.Points) AS PointsGained, AVG(points.Points) AS
AvgPointsPerGame FROM player INNER JOIN points ON player.PlayerID =
points.PlayerID INNER JOIN team ON player.Team = team.TeamName INNER
JOIN teamgames ON team.TeamName = teamgames.TeamName GROUP BY
team.TeamName, teamgames.Date;
```


✓ Showing rows 0 - 8 (9 total, Query took 0.1090 seconds.)

```
SELECT distinct team.TeamName, COUNT(teamgames.Date) AS PlayedGames,
AvgPointsPerGame FROM player INNER JOIN points ON player.PlayerID = p
INNER JOIN teamgames ON team.TeamName = teamgames.TeamName GROUP BY t
```

Show all | Number of rows: 25 Filter rows: Search this table

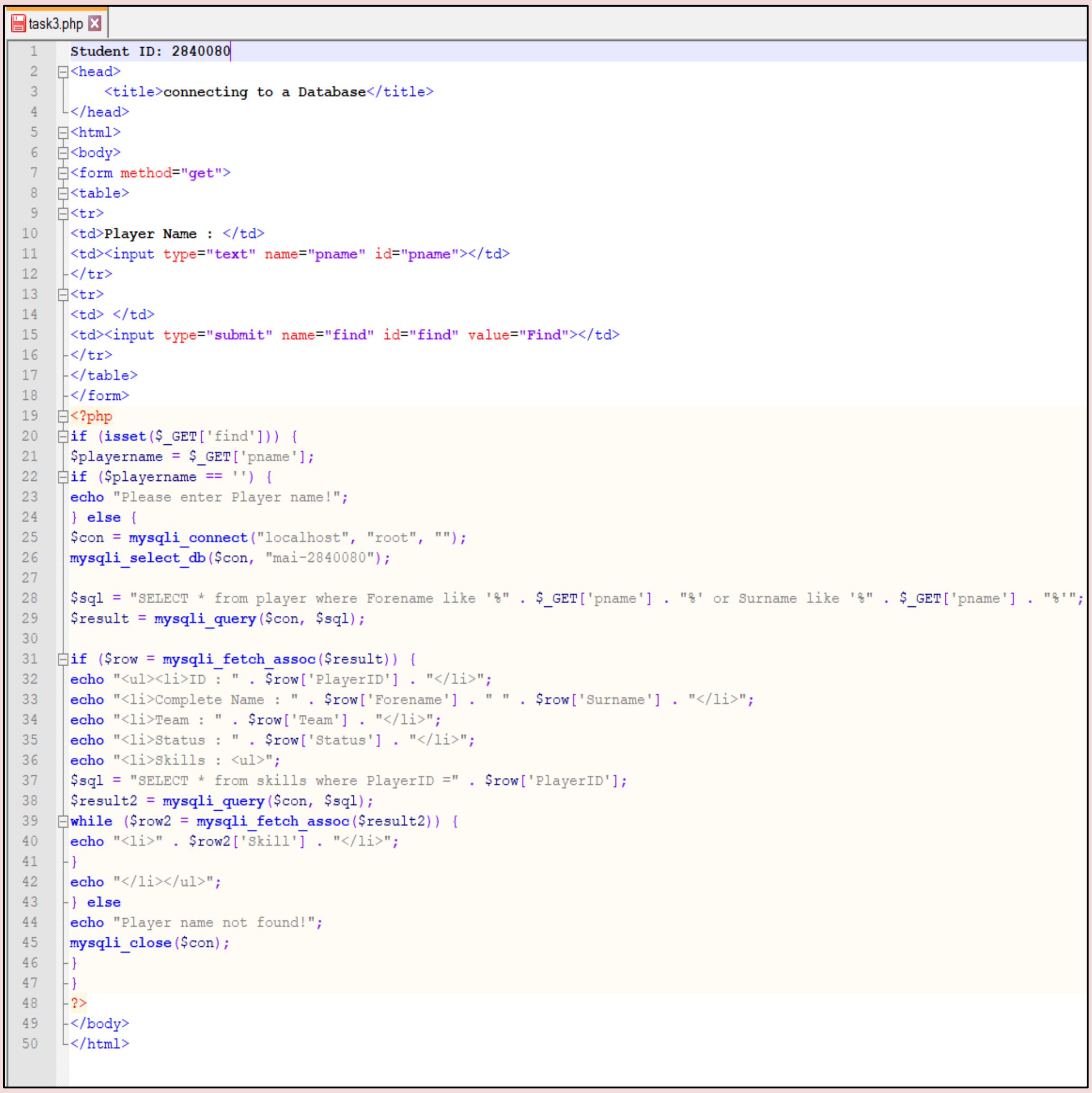
+ Options

TeamName	PlayedGames	PointsGained	AvgPointsPerGame
Blasters	182	697	3.8297
Blues	198	803	4.0556
Jets	154	598	3.8831
Racers	216	862	3.9907
Rams	100	369	3.6900
Reds	252	1051	4.1706
Rockets	95	375	3.9474
Runners	176	690	3.9205
Sharks	304	1191	3.9178

Figure 17-Result of entering the table data.

Task 3: PHP interface

This screen shows the program I used to design the PHP page interface, (I used **NotePad++).**

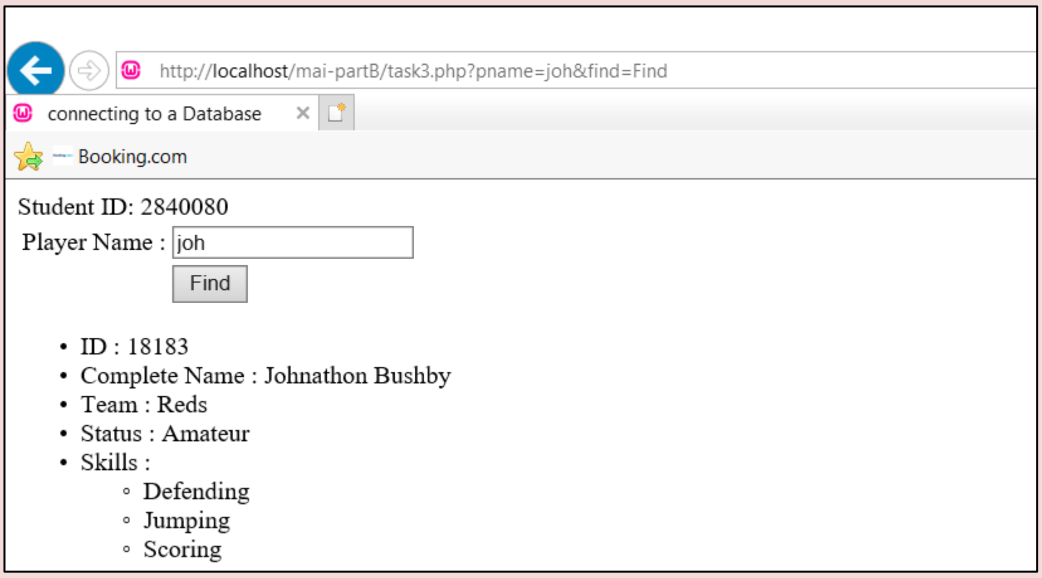


task3.php

```
Student ID: 2840080
<head>
    <title>connecting to a Database</title>
</head>
<html>
<body>
<form method="get">
<table>
<tr>
<td>Player Name : </td>
<td><input type="text" name="pname" id="pname"></td>
</tr>
<tr>
<td> </td>
<td><input type="submit" name="find" id="find" value="Find"></td>
</tr>
</table>
</form>
<?php
if (isset($_GET['find'])) {
$playername = $_GET['pname'];
if ($playername == '') {
echo "Please enter Player name!";
} else {
$con = mysqli_connect("localhost", "root", "");
mysqli_select_db($con, "mai-2840080");

$sql = "SELECT * from player where Forename like '%" . $_GET['pname'] . "%' or Surname like '%" . $_GET['pname'] . "%'";
$result = mysqli_query($con, $sql);

if ($row = mysqli_fetch_assoc($result)) {
echo "<ul><li>ID : " . $row['PlayerID'] . "</li>";
echo "<li>Complete Name : " . $row['Forename'] . " " . $row['Surname'] . "</li>";
echo "<li>Team : " . $row['Team'] . "</li>";
echo "<li>Status : " . $row['Status'] . "</li>";
echo "<li>Skills : <ul>";
$sql = "SELECT * from skills where PlayerID =" . $row['PlayerID'];
$result2 = mysqli_query($con, $sql);
while ($row2 = mysqli_fetch_assoc($result2)) {
echo "<li>" . $row2['Skill'] . "</li>";
}
echo "</li></ul>";
} else
echo "Player name not found!";
mysqli_close($con);
}
}
?>
</body>
</html>
```

Figure 18-PHP Source Code

This interface shows the Search for any player whose names (forename or surname) contain the text entered:

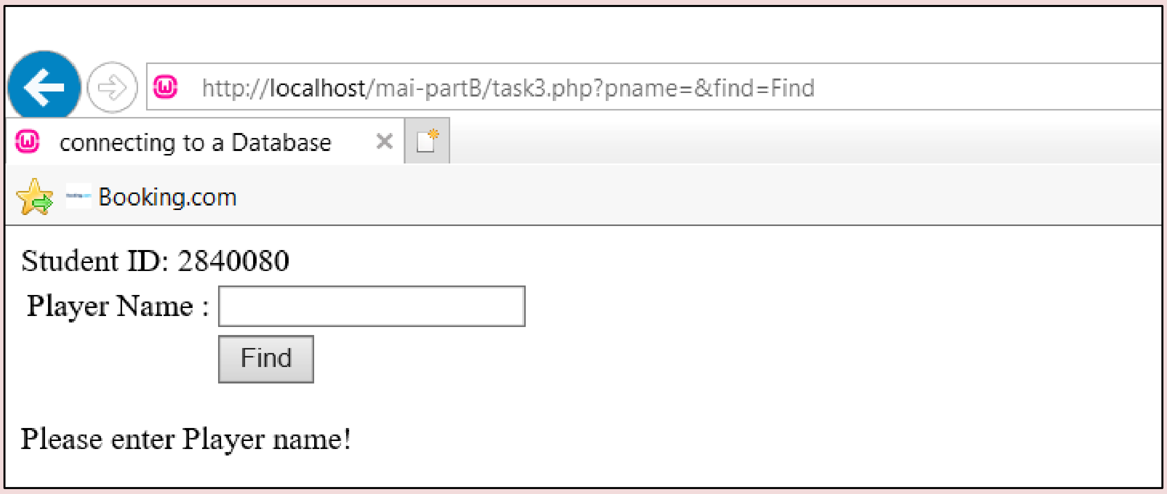


Student ID: 2840080
Player Name : joh
Find

- ID : 18183
- Complete Name : Johnathon Bushby
- Team : Reds
- Status : Amateur
- Skills :
  - Defending
  - Jumping
  - Scoring

Figure 19-result of entering the first 3 letters.

If textbox is left blank, there are a message to indicate to enter player name:

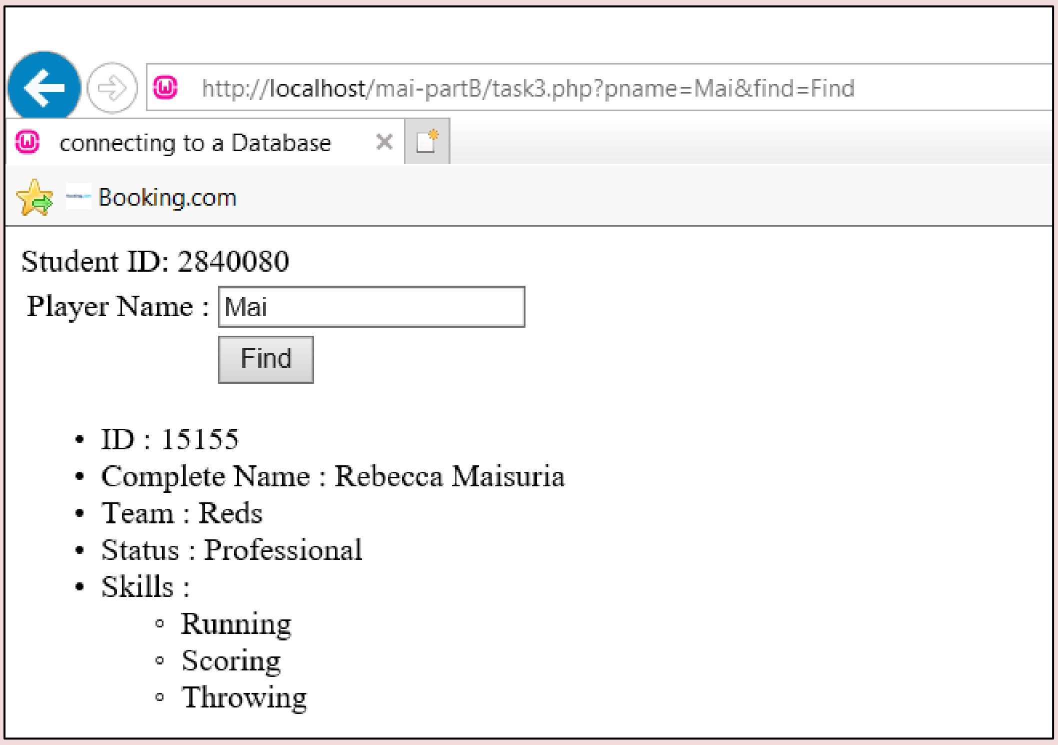


Student ID: 2840080
Player Name :
Find

Please enter Player name!

Figure 20-result of not entering a player name.

This interface shows when I enter "Mai" it searches for any name that contains these letters.



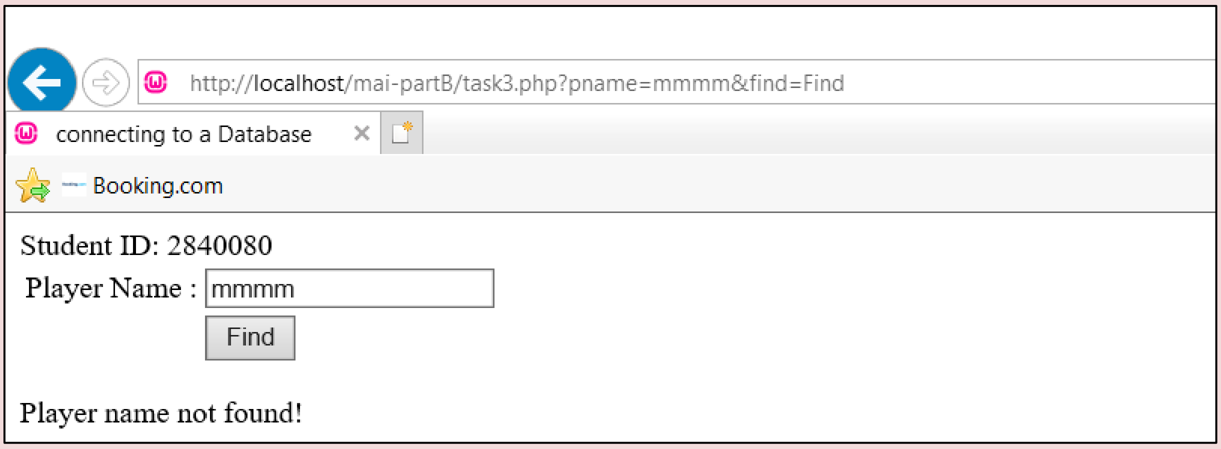
Student ID: 2840080

Player Name : Mai

Find

- ID : 15155
- Complete Name : Rebecca Maisuria
- Team : Reds
- Status : Professional
- Skills :
  - Running
  - Scoring
  - Throwing

This interface shows that database do not find a player whose name contains the text "mmmm":



Student ID: 2840080

Player Name : mmmm

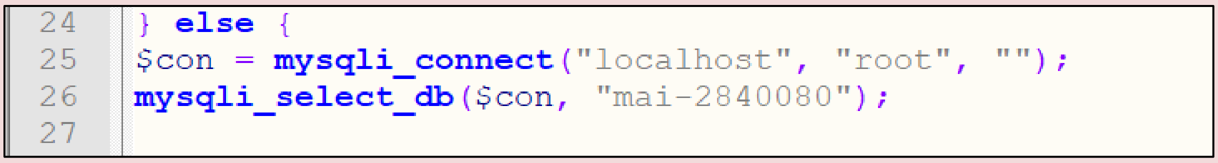
Find

Player name not found!

Figure 21-result of entering a wrong name.

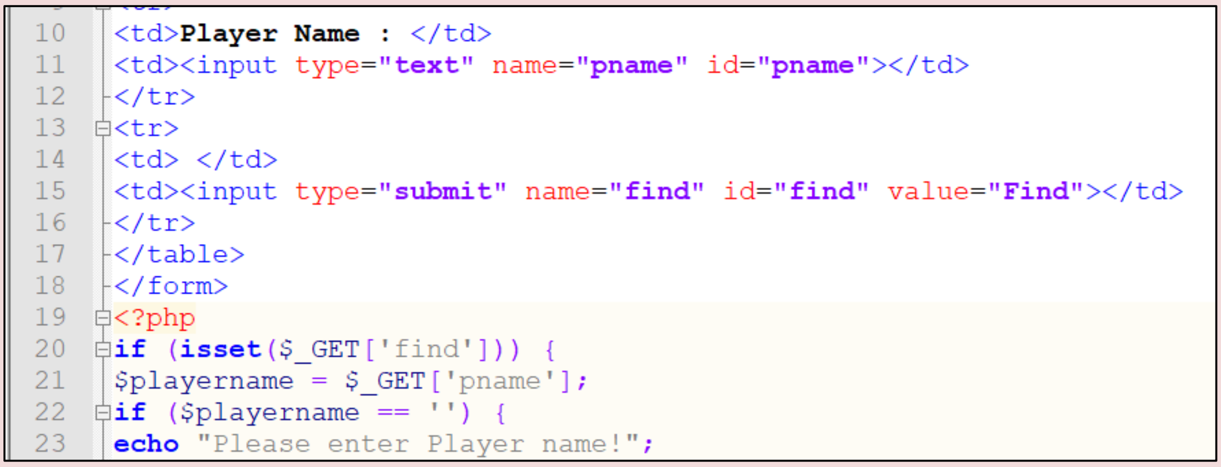
PHP Code:

```
<head>

     <title>connecting to a Database</title>

</head>

<html>

<body>

<form method="get">

<table>

<tr>

<td>Player Name : </td>

<td><input type="text" name="pname" id="pname"></td>

</tr>

<tr>

<td> </td>

<td><input type="submit" name="find" id="find"
value="Find"></td>

</tr>

</table>

</form>

<?php

if (isset($_GET['find'])) {

$playername = $_GET['pname'];

if ($playername == '') {

echo "Please enter Player name!";

} else {

$con = mysqli_connect("localhost", "root", "");

mysqli_select_db($con, "mai-2840080");
$sql = "SELECT * from player where Forename like '%" .
$_GET['pname'] . "%' or Surname like '%" . $_GET['pname'] .
"%'";

$result = mysqli_query($con, $sql);



if ($row = mysqli_fetch_assoc($result)) {

echo "<ul><li>ID : " . $row['PlayerID'] . "</li>";

echo "<li>Complete Name : " . $row['Forename'] . " " .
$row['Surname'] . "</li>";

echo "<li>Team : " . $row['Team'] . "</li>";

echo "<li>Status : " . $row['Status'] . "</li>";

echo "<li>Skills : <ul>";

$sql = "SELECT * from skills where PlayerID =" .
$row['PlayerID'];

$result2 = mysqli_query($con, $sql);

while ($row2 = mysqli_fetch_assoc($result2)) {

echo "<li>" . $row2['Skill'] . "</li>";

}

echo "</li></ul>";

} else

echo "Player name not found!";

mysqli_close($con);

}

}

?>

</body>

</html>
```



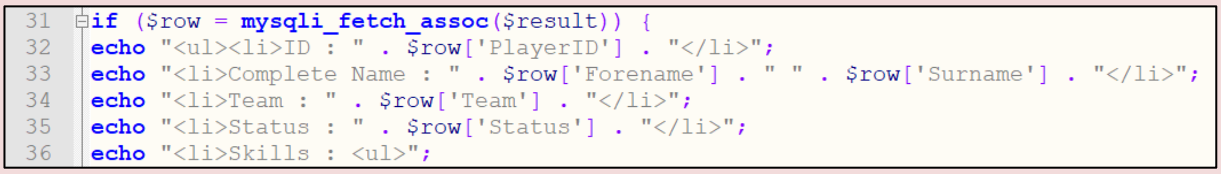
```
} else {
$con = mysqli_connect("localhost", "root", "");
mysqli_select_db($con, "mai-2840080");

```

Figure 22



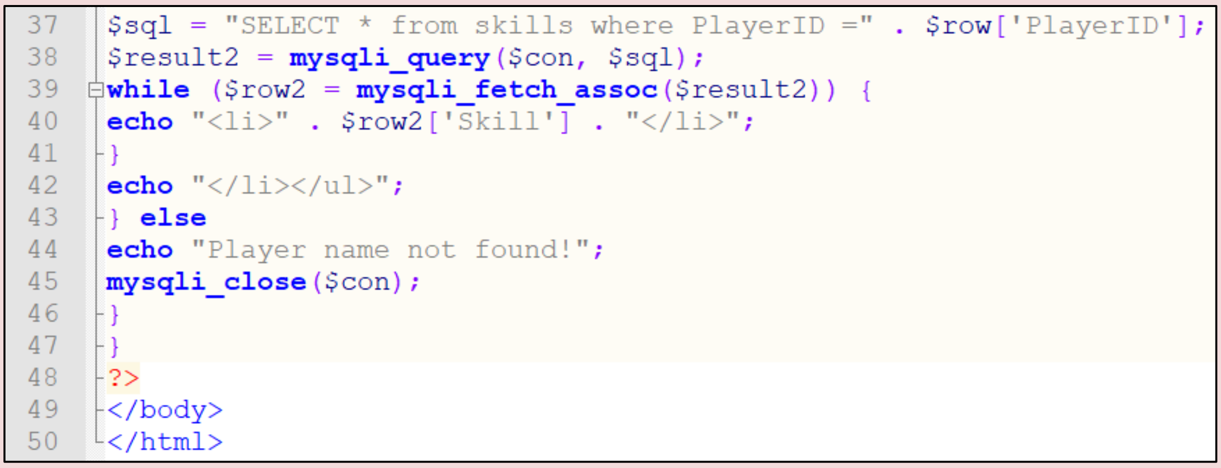
```
<td>Player Name : </td>
<td><input type="text" name="pname" id="pname"></td>
</tr>
<tr>
<td> </td>
<td><input type="submit" name="find" id="find" value="Find"></td>
</tr>
</table>
</form>
<?php
if (isset($_GET['find'])) {
$playername = $_GET['pname'];
if ($playername == '') {
echo "Please enter Player name!";
```

Figure 23



```
if ($row = mysqli_fetch_assoc($result)) {
echo "<ul><li>ID : " . $row['PlayerID'] . "</li>";
echo "<li>Complete Name : " . $row['Forename'] . " " . $row['Surname'] . "</li>";
echo "<li>Team : " . $row['Team'] . "</li>";
echo "<li>Status : " . $row['Status'] . "</li>";
echo "<li>Skills : <ul>";
```

Figure 24



```
$sql = "SELECT * from skills where PlayerID =" . $row['PlayerID'];
$result2 = mysqli_query($con, $sql);
while ($row2 = mysqli_fetch_assoc($result2)) {
echo "<li>" . $row2['Skill'] . "</li>";
}
echo "</li></ul>";
} else
echo "Player name not found!";
mysqli_close($con);
}
}
?>
</body>
</html>
```

Figure 25

Explanation about php

I created queries to searching the players whose names if for first name or last name, and showing in the php interface, where the player properties are displayed by simply typing the name, last name or one of the letters for the players name, which will display the following properties : \`PlayerID\`, \`Forename\`, \`Surname\`, \`Team\`, \`Status\`, \`Skills\`.

The reason for creating this is for one of the queries to search in the SQL table and take all the required data and at the same time the other query searches for another table in Mai-2840080 and collects all the skills of each player in one row and then display them with each player.

End.