ASSIGNMENT

Designing and creating a Relational Database 2018

CSCU9B3

Computing Science and Maths

University of Stirling

STUDENT NUMBER : 2636568

Contents

CSCU9B3 Relational Database Assignment 2018 Computing Science and Maths, University of Stirling 40% of module grade; due 4pm 19th November

Each week in the Stirling area, sporting teams meet to compete in games. More than two teams can compete in any given game and players win points for their team rather than winning a game outright. Each week of the year there is one game, which takes place at one of a few possible venues. Not all teams compete every week. Teams can have different numbers of players. At the end of the year, the average points scored by each team are calculated and a league table is created.

The data from the games, including team members and other information is stored in a very badly designed way. One long file is kept and at the end of each game, the points scored by each player are added to the file. The problem is that all the information about the player, the team and the game has to be included in every row of the file, so there is massive data duplication.

The following data is stored about each player:

- • A unique ID
- • Forename
- • Surname
- • Team name
- • Status (Professional or Amateur)
- • Skills that they possess

This is stored about each team:

- • Team name
- • Home town

All of this data is stored in each row of the file, along with the following data about a game:

- • The date of the game
- • The venue of the game (where it was played)
- • The number of points scored by the player

For example:

What is worse, if a player has more than one skill, then the same data is repeated for each of the skills, as you can see above. The data does not explicitly list every team that took part in each game: to see that you would need to look at every row for a given date. Note that there is only one event on any given date.

It is your job to turn this data into a relational database.
The data is stored in the rawdata.csv file (and a version without header information, for loading into the database, in noheaddata.csv), which you can download from the module's Canvas assignment page.

Your assignment is to complete the following steps and present your results in a written report. These instructions are detailed and following them properly should

ensure

you get good marks:

1. Design a set of tables for a relational database to store this data.
2. In your report give an ER diagram showing the relationships between the

   tables. In your diagram, make sure you:

   1. Put the table name at the top of each table
   2. List the fields in each table
   3. Underline the primary key fields
   4. Put a * after the foreign key fields
   5. Mark the cardinality of each relationship at both ends of the

      connecting line

   6. Indicate optionality with a dashed line.
3. In your report, write a justification for your design, considering aspects such as data integrity and normalisation.
4. Create these tables in MySQL by writing and executing (via the phpMyAdmin interface) the SQL for creating each of the tables, including all primary and foreign key definitions. Make sure you choose sensible types for the fields. In your report, show the SQL you have used.
5. Write and execute SQL to create a table to hold the data from the nohead.csv file and then upload the data into your database via the phpMyAdmin "import" facility (no need to mention this step in the report).
6. Use SQL statements to extract the data from your first table into the correct tables that you created above. In your report, show the SQL for doing this for one example table only.
7. In your report, write the SQL you would use to answer each of the following questions, and also include the results you get from executing the query. You must not use the table from step 5 – make all your queries from the tables corresponding to your ER decomposition made in step 1.
   1. List each team name and the town they are based in.
   2. List the total number of games played by each team.
   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).

   4. List the dates of all the games where the Jets and the Rams both

      played.

   5. Write a query to 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.

8. Starting with the template file, assignment.php (available from Canvas), using the PHP mysqli package (either the procedural or object-oriented version) complete the PHP and SQL required to take whatever text is entered in the form box and do the following:

1. Search for any players whose names (forename or surname) contain the text entered.
2. Display neatly in the web page the following characteristics of all matching players found: ID, forename, surname, team, status, skills

3. Try out your code by placing this file (DO NOT rename it) in your web folder on wamp0, as you did in practical exercises (\\wamp0.cs.stir.ac.uk\www\xxx where xxx is your username).
4. In your report, include a copy of your code (please remove your password and any other sensitive information from the copy shown in the report) and give its URL ie http://wamp0.cs.stir.ac.uk/xxx/assignment.php.

Make sure your code is robust against any mistakes or malicious intent in text entered in the form box by a user of your webpage. During marking, your webpage will be trialled and this will be checked. As above, you should carry out your queries on the tables corresponding to your ER decomposition in part 1.

Marking breakdown and criteria

Your work will be marked out of 100, according to the following breakdown:

1. ER diagram: 20%
2. Justification of design: 20%
3. Table creation: 10%
4. Data transfer: 5%
5. Searches: 25%
6. PHP (Assignment.php): 15%
7. Overall quality of your report: 5%

Submitting your work and assessment procedures

The assignment will be submitted as an electronic (PDF) type-written report uploaded (via Turnitin) to Canvas by 4pm on Monday 19th November. DO NOT put your name in the report, only your registration number.

In the report, include all the components listed in the assignment steps. The report should be professionally presented and easy to read. 5% of the marks will be given for the quality of the report.

Submission also includes placing your completed file, assignment.php, in your web- accessible folder on wamp0: \\wamp0.cs.stir.ac.uk\www\xxx where xxx is your username.

Late submission

It is possible for the co-ordinator to vary the dates for a student. If you must submit your work late (perhaps for medical reasons) then this can be arranged, but it must be discussed with the co-ordinator as soon as possible after the problem becomes known. Assessed coursework submitted late will be accepted up to seven calendar days after the submission date (or expiry of any agreed extension) but the grade will be lowered by three marks per day or part thereof. After seven days the piece of work will be deemed a non-submission and will result in a fail grade for the module as a whole.

Plagiarism

Work which is submitted for assessment must be your own work. All students should note that the University has a formal policy on plagiarism which can be found at:

https://www.stir.ac.uk/about/faculties-and-services/academic-registry/academic- policy-and-practice/quality-handbook/assessment-and-academic-misconduct/#eight

How marks will be awarded

Marks will be awarded both for the technical correctness of what you have done and also for the clarity and organisation of how you describe it (this does not mean that you should give a long account of what you did).

The assignment counts for 40% of the total course mark.
The deadline for submission is Monday 19th of November 2018 at 4pm.

DRAWING ER DIAGRAM

According to the table, it is the **first normal form relation** because there are many duplications for the data. Before to second normal form, we should set up the relation between attributes and set up primary keys and foreign key.

Functional Dependency relation (one to one relation)

1. Student ID → Forename (one student id identify one forename)
2. Forename → Surname.
3. Student ID → team.
4. Student ID → Status.
5. Student ID, skills → Points.
6. Team Name → Hometown
7. Date → Venue
8. Student ID → Hometown.

Functional Dependency relation (many to many relation)

1. Student ID → Skills.
2. Skills → team.
3. Skills. → Date.

Designing 2NF ER models(Considering by primary keys)

Students
1.Student ID
2.Forename
3.Surname
4.team
5.Status
6.Hometown

Skills
1.Student ID
2.Skills
3.team
4.Date
5.Venue
6.Point

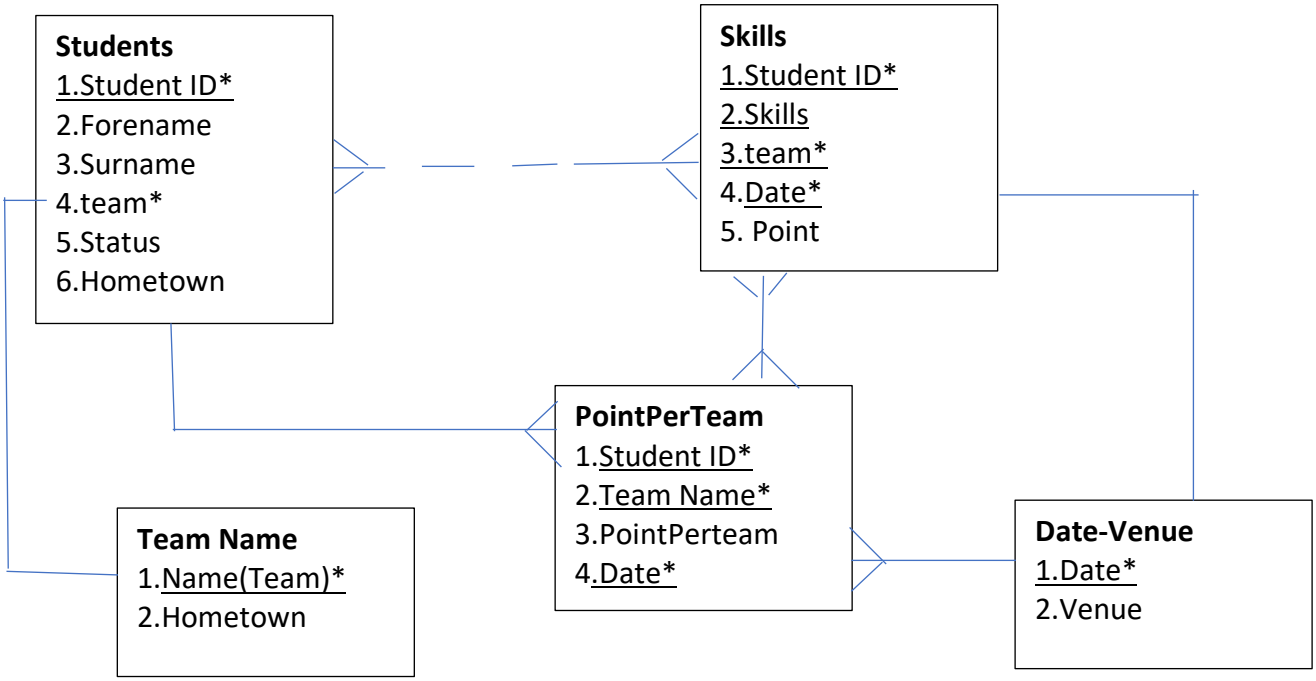
Students(Student ID,Forename,Surname,team,status,hometown)

Skills(Student ID, Skills, team,Date,Venue,Point)

Now we set the **student ID to be primary keys** and foreign keys because of linking between Skills table. For Skills table, storing detail for student, **primary keys** are Student ID, Skills and team .

Designing 3NF ER models

Considering the second normal form, if we want to design the third normal form. We have to find **transitive attributes** both tables. We find **Team Name** → **Hometown**, **Date** → **Venue** and we want the table for the **point per team** because the skills team show the point for each student per team.



The tables consist of

1.Students(Student ID*,Forename,Surname,team*,status,hometown)

2.Skills(Student ID*, Skills, team*,Date*, ,Point)

3. Team Name(Name*,Hometown)

4.Date-Venue(Date*,Venue)

5.PointPerTeam(Student ID*,Team Name*,PointPerTeam,Date*)

So the relation between Students and Skills is **many to many**, but one students didn't play all sports so we use **dashed line** representing the relation. **However**, all teams have to play all sports and all sports are played by players in all teams so these relations are many to many. **And** we have **one to one** function between team Name and Students table because one student is in none team and one team has one hometown. **Also** the date of game has been in one Venue so this relation will be **one to one**. **Finally** we have two new tables. Those primary keys are Name (Team) and Date, including foreign keys to Date and Name for accessing to other tables.For **PointPerteam** table,all point are

given by all players so there are not optional. From **one to many** relation,one student have many team score and one date have many scores.

SQL Statements
(Table creation and Data transfer)

Before we import the **noheaddata.csv** file, we should create the table by writing SQL like Example1.

Example1 (noheaddataB3 table)

```
CREATE TABLE `tis00024`.`noheaddataB3` ( `Student ID` INT(5) , `Forename` VARCHAR(11), `Surname` VARCHAR(12), `Team name` VARCHAR(9), `Status` VARCHAR(12), `Skills` VARCHAR(10) , `Name of team` VARCHAR(9), `Home town` VARCHAR(20), `Venue` VARCHAR(20) , `Date` DATE, `Point` INT(1));
```

After we import the data into **noheaddataB3 table.** We have to create another tables that we use ER model representing the tables. Then we have to extract the data from **noheaddataB3 table** .Look at the Example2.

Example2 (Import data to Students table)

```
INSERT INTO `students`(`Student ID`,`Forename`, `Surname`, `Status`, `Team Name`) SELECT DISTINCT `Student ID`,`Forename`, `Surname`, `Status`, `Team name ` FROM noheaddatab3;
```

Before importing, we should create another table following examples.

Example3 (Creating `students` table)

```
CREATE TABLE `tis00024`.`students` ( `Student ID` INT(5),`Forename` VARCHAR(12), `Surname` VARCHAR(16), `Status` VARCHAR(12), `Team Name` VARCHAR(8)) ;
```

Example4 (Creating `team name` table)

```
CREATE TABLE `tis00024`.`Team name` ( Name` VARCHAR(10),`Home town` VARCHAR(18)) ;
```

Example5 (Creating `games` table)

```
CREATE TABLE `tis00024`.`Games` ( `Student ID` INT(5),`Team` VARCHAR(10)  , `Skills` VARCHAR(11), `Point` INT(1), `Date` DATE);
```

Example6 (Creating `date-venue` table)

```
CREATE TABLE `tis00024`.`Date-Venue` ( `Date of game` DATE , `Venue` VARCHAR(20));
```

Example7 (Creating `pointperteam` table)

```
CREATE TABLE `tis00024`.`Pointpergames` ( `Student ID` INT(5),`Team Name` VARCHAR(8),`Skills` VARCHAR(11),`PointPerTeam` INT(1),`Date`Date) ;
```

Then we set up primary keys to `**Student ID**` from students and games tables. In games table, we have to set up primary keys by setting (**Student ID, team,Date,Skills**) in the same time for protect duplication of data. And we use `**Name**` from team name table and `**Date of game**` from date-venue table to be primary keys.

Foreign Keys (Using Relation view **from structure in phpMyAdmin)**

1.From students table to games table and from games to students table.

"Student ID" → "Student ID"

2. From students table to team name table .

"Team Name" → "Name"

3.From Pointperteam table to games table(we want to know about detail of competition each team) and from games to Pointperteam table.

"Team Name" → "Team"

"Team" → "Team Name"

4.from games to date-venue table or from date-venue table to games table.

"Date" → "Date of game"

"Date of game" → "Date"

5.from Pointperteam to Students table and PointPerteam to Date-venue table

"Student ID" → "Student ID"

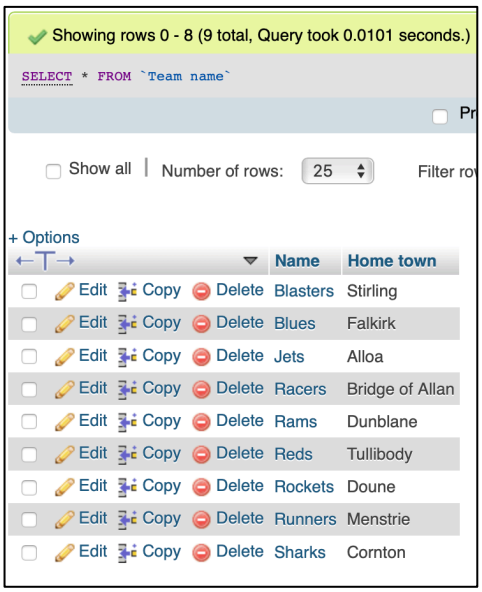
"Date" → "Date of game"

Query Question

1. List each team name and the town they are based in.

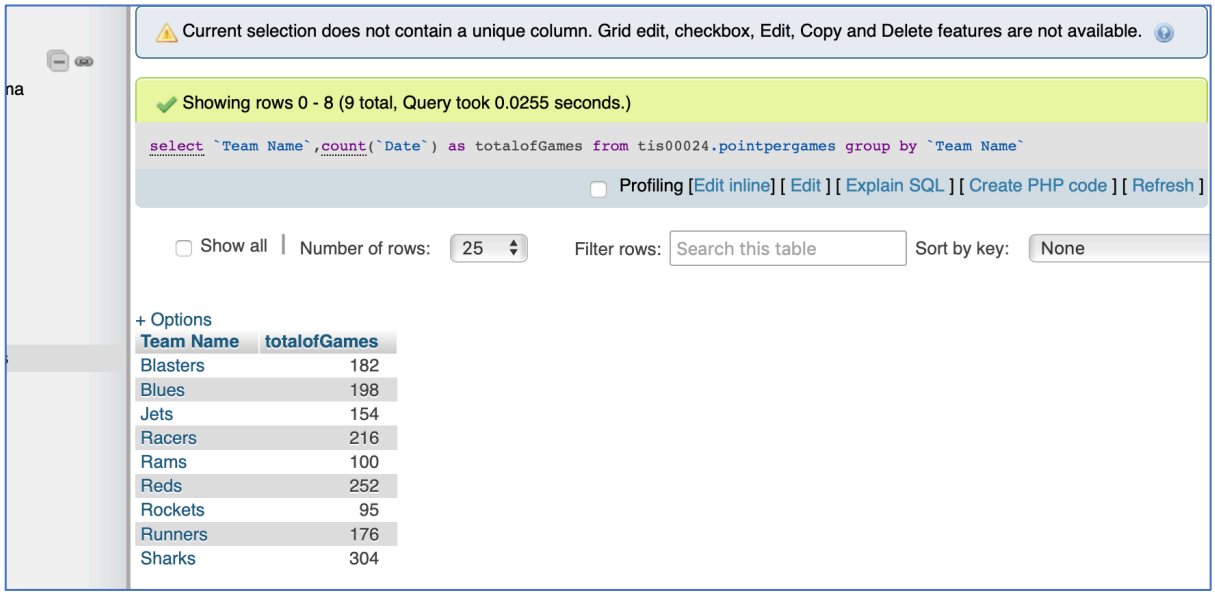
```
SELECT * FROM `Team name`;
```

result



2. List the total number of games played by each team.

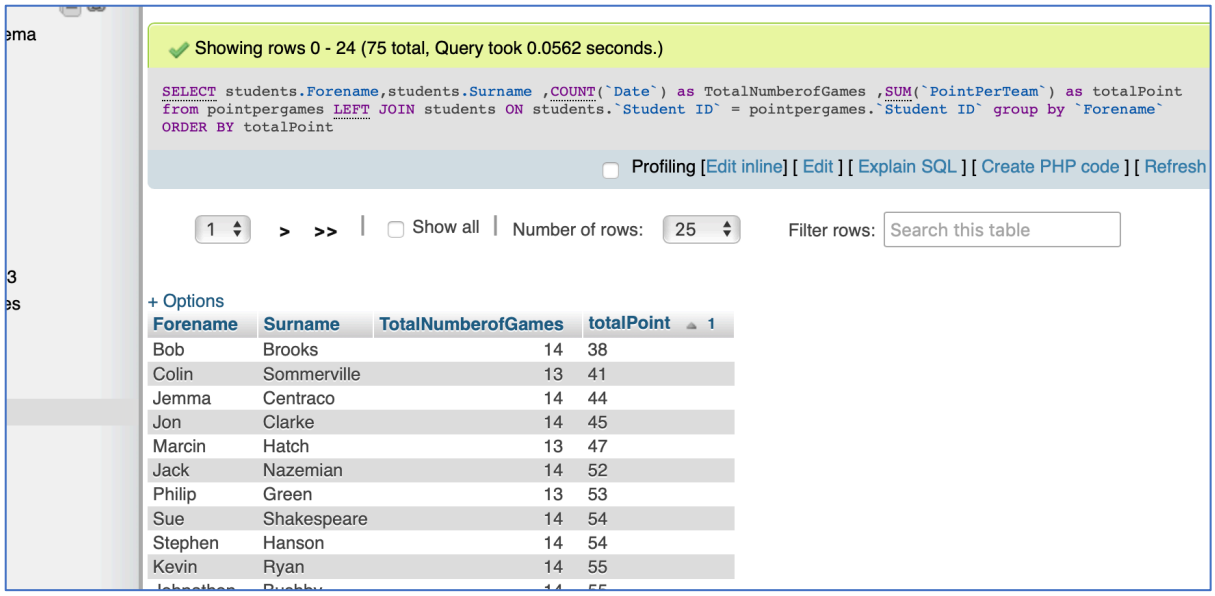
```
select `Team Name`,count(`Date`) as totalofGames  from tis00024.games
group by `Team Name` ;
```



Team Name	totalofGames
Blasters	182
Blues	198
Jets	154
Racers	216
Rams	100
Reds	252
Rockets	95
Runners	176
Sharks	304

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  students.Forename,students.Surname ,COUNT(`Date`) as
TotalNumberofGames ,SUM(`PointPerTeam`) as totalPoint   from
pointpergames  LEFT JOIN students ON students.`Student ID` =
pointpergames.`Student ID` group by `Forename`
```



Showing rows 0 - 24 (75 total, Query took 0.0562 seconds.)

```
SELECT students.Forename,students.Surname ,COUNT(`Date`) as TotalNumberofGames ,SUM(`PointPerTeam`) as totalPoint
from pointpergames LEFT JOIN students ON students.`Student ID` = pointpergames.`Student ID` group by `Forename`
ORDER BY totalPoint
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

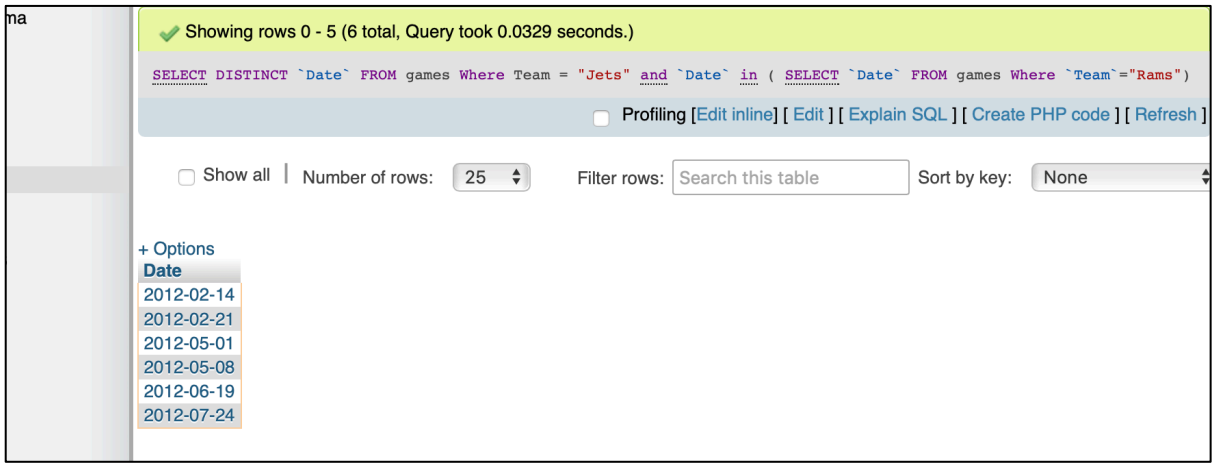
1 > >> | Show all | Number of rows: 25 Filter rows: Search this table

+ Options

Forename	Surname	TotalNumberofGames	totalPoint ▲ 1
Bob	Brooks	14	38
Colin	Sommerville	13	41
Jemma	Centraco	14	44
Jon	Clarke	14	45
Marcin	Hatch	13	47
Jack	Nazemian	14	52
Philip	Green	13	53
Sue	Shakespeare	14	54
Stephen	Hanson	14	54
Kevin	Ryan	14	55

4.List the dates of all the games where the Jets and the Rams both played.

```
SELECT DISTINCT `Date`FROM games
Where Team = "Jets" and `Date` in ( SELECT `Date` FROM games Where
`Team`="Rams");
```



Showing rows 0 - 5 (6 total, Query took 0.0329 seconds.)

```
SELECT DISTINCT `Date` FROM games Where Team = "Jets" and `Date` in ( SELECT `Date` FROM games Where `Team`="Rams")
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

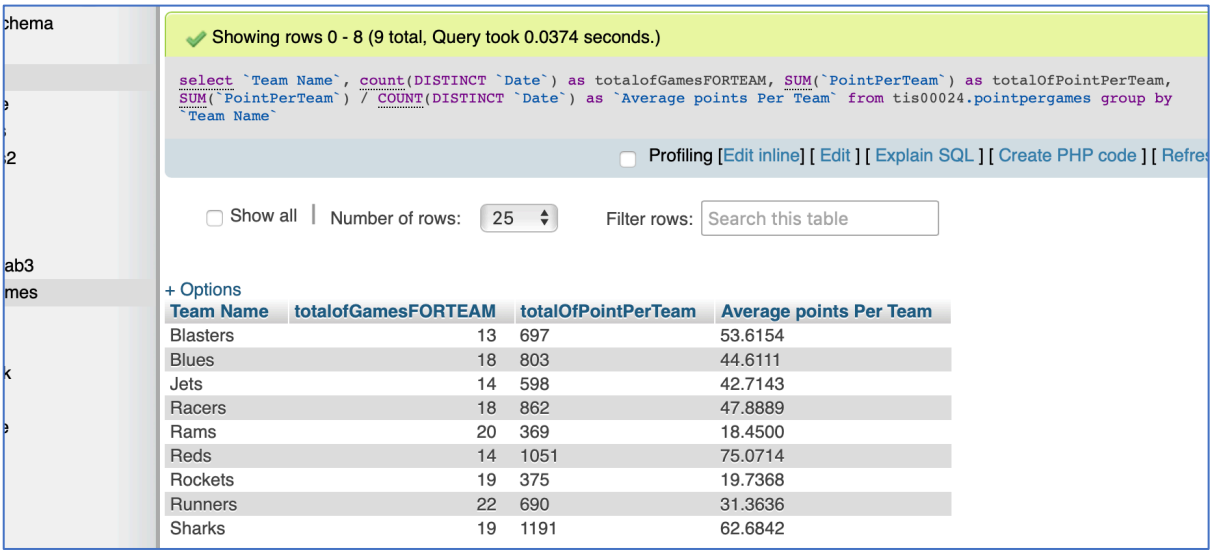
+ Options

Date
2012-02-14
2012-02-21
2012-05-01
2012-05-08
2012-06-19
2012-07-24

5.Write a query to 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 `Team Name`, count(DISTINCT `Date`) as totalofGamesFORTEAM,
SUM(`PointPerTeam`) as totalOfPointPerTeam, SUM(`PointPerTeam`) /
COUNT(DISTINCT `Date`) as `Average points Per Team`
from tis00024.pointpergames  group by `Team Name`
```

Result from question 5



Showing rows 0 - 8 (9 total, Query took 0.0374 seconds.)

```
select `Team Name`, count(DISTINCT `Date`) as totalofGamesFORTEAM, SUM(`PointPerTeam`) as totalOfPointPerTeam,
SUM(`PointPerTeam`) / COUNT(DISTINCT `Date`) as `Average points Per Team` from tis00024.pointpergames group by
`Team Name`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refres

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

+ Options

Team Name	totalofGamesFORTEAM	totalOfPointPerTeam	Average points Per Team
Blasters	13	697	53.6154
Blues	18	803	44.6111
Jets	14	598	42.7143
Racers	18	862	47.8889
Rams	20	369	18.4500
Reds	14	1051	75.0714
Rockets	19	375	19.7368
Runners	22	690	31.3636
Sharks	19	1191	62.6842

PHP template files

From

http://wamp0.cs.stir.ac.uk/tis00024/assignment.php.

```
<html>
<head>
        <title>Assignment Template</title>
</head>

<body>
  <form action ="assignment.php" method="post">
                Search player name: <input type="text" name="name">
                <input type="submit">
  </form>
```

```
<?php
$servername = "wamp0.cs.stir.ac.uk";
$username = "xxx";
$password = "xxx";
$database = "xxx";

if(!empty($_POST['name'])) {

	$name = $_POST['name'];
	echo "Welcome to sporting games in Stirling Area,It 's nice to meet you ,
{$_POST["name"]}, .<br>";

	// Create connection
	$conn = mysqli_connect($servername, $username, $password,
$database);

	// Check connection
	if (!$conn) {
	   die("Connection failed: " . mysqli_connect_error());
	}
	echo "Connected successfully to database.<br>";

	$name=mysqli_real_escape_string($conn,$name);
	$name=strip_tags($name);

	$sql = "SELECT DISTINCT students.`Student
ID`,`Forename`,`Surname`,`Status`,`Team Name`,games.Skills FROM students
, games  WHERE Forename LIKE '%".$name."%' OR Surname LIKE
'%".$name."%' ";
	echo "Query is: ".$sql."<br>";


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

	if (!$result) {
		echo "Search produced an error: ". mysqli_error($conn);
	}
	else {
		echo "Information/History:<br>";
	// output data of each row
	while($row = mysqli_fetch_row($result)) {
```

```
        echo "<br>Student ID: " . $row[0]. " <br>Forename : " . $row[1]. "
<br>Surname : " . $row[2]. " <br>Status : " . $row[3]." <br>Team Name : " .
$row[4]." <br>Skills : " . $row[5]."<br>";

                }
}
mysqli_close($conn);
}


?>
</body>
</html>
```