**CSCU9B3 ASSIGNMENT**

**STUDENT NUMBER - 2615649**

1. Design a set of tables for a relational database to store this data.

After analysing the problem I found the following tables best represent the data in the format - table name: field names

Player: ID, Forename(VARCHAR), Surname(VARCHAR)

Skill: ID\*(INT1), Status(VARCHAR), Skill(VARCHAR)

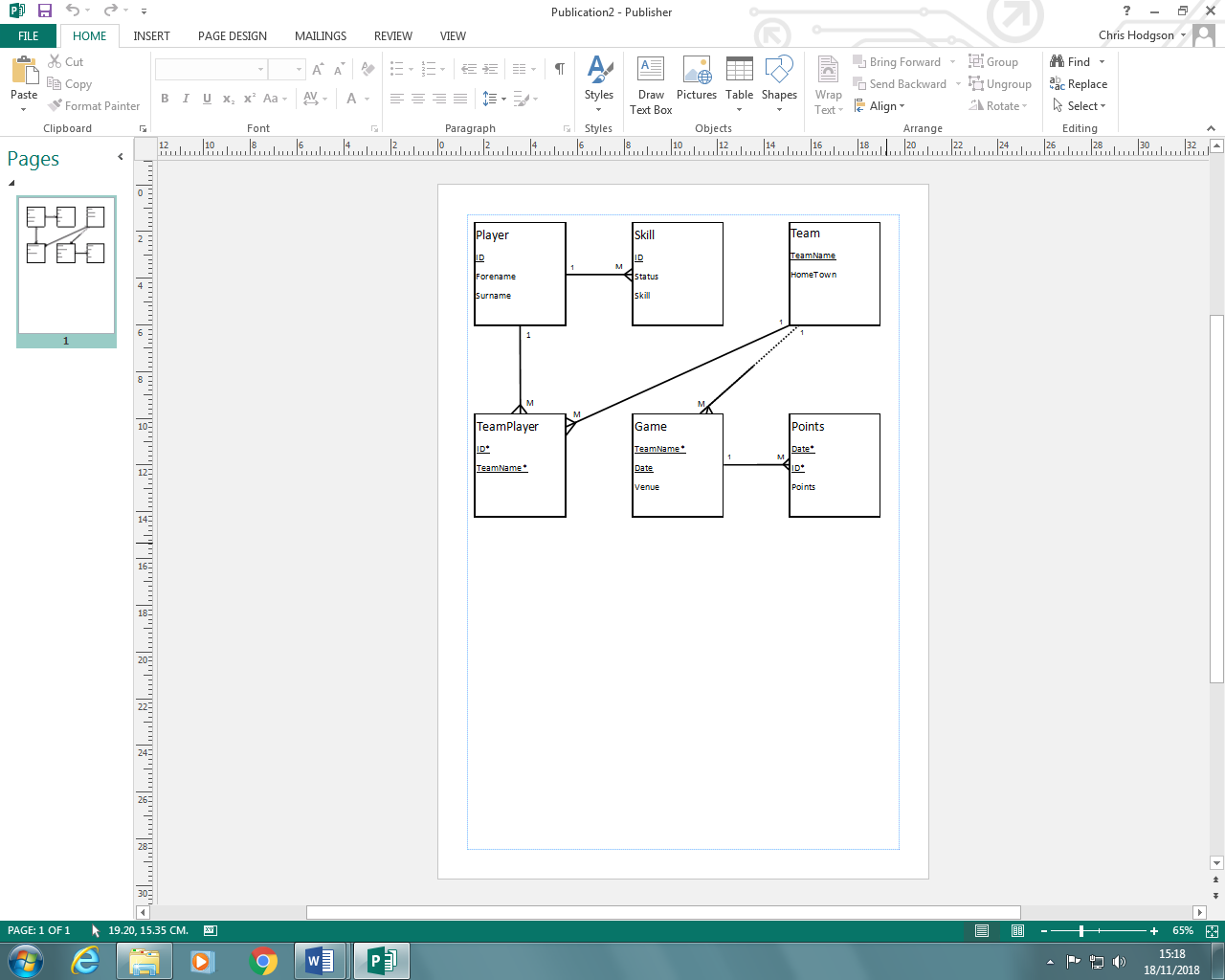
Team: TeamName(VARCHAR), HomeTown(VARCHAR)

TeamPlayer: ID\*(INT1), TeamName\*(VARCHAR)

Game: TeamName\*(VARCHAR), Date(DATE), Venue(VARCHAR)

Points: Date\*(DATE), ID\*(INT), Points(INT)

1 could be VARCHAR as not used for calculations.



1. Justification of design

Data integrity

Entity integrity

* Every table has a primary key where the values are unique.
* Each row in the database is unique to the furthest extent possible.
* Data is intact and unchanged

Referential integrity

* All foreign keys have matching primary key values in another table.
* All data in the table is connected to the necessary data. This means that the database is searchable and recoverable.
* A benefit of this is increased maintainability and stability in the system.

Domain integrity

* All columns have the appropriate data types in the database.

Normalization

First normal form

* Contains tables with rows and columns
* All entries in each column are the same data type
* Every row is unique
* All columns have unique names

Second normal form

* Is in first normal form
* All non-key attributes are fully functionally dependant on the primary key

Third normal form

* Is in second normal form
* No transitive functional dependencies.

Data types justification

* ID INT PRIMARY KEY, INT because no values begin with 0 so its suitable.
* Forename VARCHAR(50), text field no longer than 50 characters to allow for long names up to 50 characters. Requires less storage CHAR
* Surname VARCHAR(50), “ ”
* TeamName VARCHAR(50), “ ”
* `Status` VARCHAR(50), “ ”
* Skill VARCHAR(50), “ ”
* TeamName2 VARCHAR(50), “ ”
* HomeTown VARCHAR(50), “ ”
* Venue VARCHAR(50), “ ”
* `Date` DATE, a date is being represented in the format YYY-MM-DD
* Points INT, number being used for calculations so int is appropriate

CREATE TABLE IF NOT EXISTS`Player`

(

ID INT PRIMARY KEY,

Forename VARCHAR(50),

Surname VARCHAR(50)

);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CREATE TABLE IF NOT EXISTS`Skill`

(

ID INT PRIMARY KEY,

`Status` VARCHAR(50),

Skill VARCHAR(50),

FOREIGN KEY (ID) REFERENCES Player(ID)

);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CREATE TABLE IF NOT EXISTS`team`

(

TeamName VARCHAR(50) PRIMARY KEY,

HomeTown VARCHAR(50)

);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CREATE TABLE IF NOT EXISTS`teamplayers`

(

ID INT,

TeamName VARCHAR(50),

PRIMARY KEY(ID, TeamName),

FOREIGN KEY (ID) REFERENCES Plyaer(ID)

FOREIGN KEY (TeamName) REFERENCES Team(TeamName)

);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CREATE TABLE IF NOT EXISTS`Game`

(

TeamName VARCHAR(50),

`Date` DATE,

Venue VARCHAR(50),

PRIMARY KEY (TeamName, `Date`)

);

ALTER TABLE Game

ADD FOREIGN KEY (TeamName) REFERENCES team(TeamName);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CREATE TABLE IF NOT EXISTS`GamePoints`

(

`Date` DATE,

ID INT,

Points INT,

PRIMARY KEY (`Date`, ID),

FOREIGN KEY (ID) REFERENCES Player(ID)

);

ALTER TABLE GamePoints

ADD FOREIGN KEY (Date) REFERENCES game(Date);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Data imported into temp table

CREATE TABLE IF NOT EXISTS`tempload`

(

ID INT PRIMARY KEY,

Forename VARCHAR(50),

Surname VARCHAR(50),

TeamName VARCHAR(50),

`Status` VARCHAR(50),

Skill VARCHAR(50),

TeamName2 VARCHAR(50),

HomeTown VARCHAR(50),

Venue VARCHAR(50),

`Date` DATE,

Points INT

);

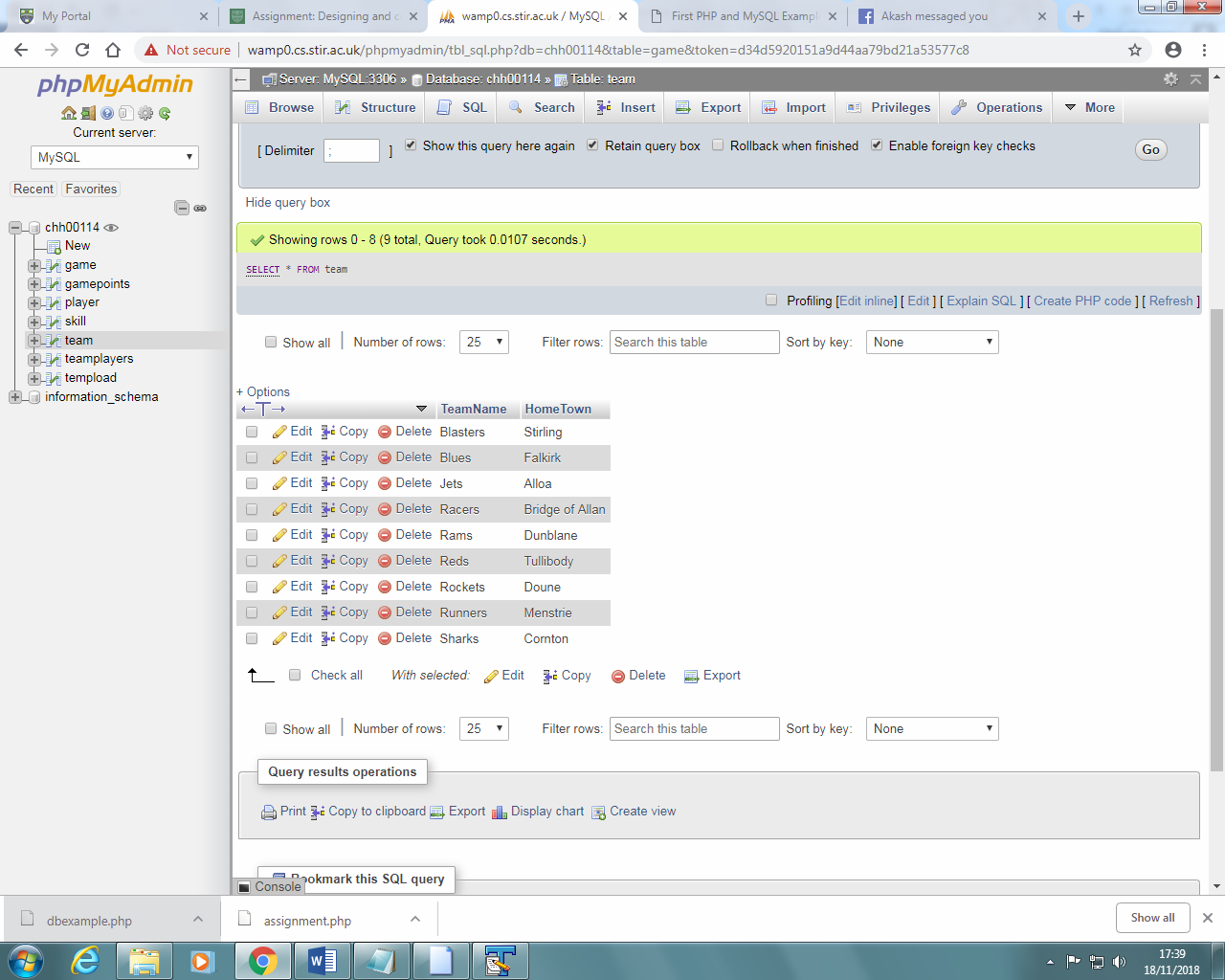
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

INSERT INTO Player (ID, ForeName, Surname)

SELECT DISTINCT ID, ForeName, Surname FROM tempload

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

SELECT \* FROM team



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

SELECT COUNT(Date), TeamName

FROM game

GROUP BY TeamName



1. 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 gamepoints.ID, player.Forename, COUNT(gamepoints.Date) AS GamesPlayed, SUM(gamepoints.Points) AS PlayerPoints FROM gamepoints, player WHERE player.ID = gamepoints.ID GROUP BY gamepoints.ID



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

SELECT game.Date

FROM game

WHERE TeamName IN ('Jets', 'Rams')

GROUP BY game.Date

HAVING COUNT(DISTINCT game.TeamName) = 2



1. 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.TeamName,

COUNT(DISTINCT gamepoints.Date) AS GamesPlayed,

(SELECT SUM(gamepoints.Points)) AS TotalPoints,

(SELECT SUM(gamepoints.Points)/COUNT(DISTINCT gamepoints.Date)) AS TeamsAveragePoints

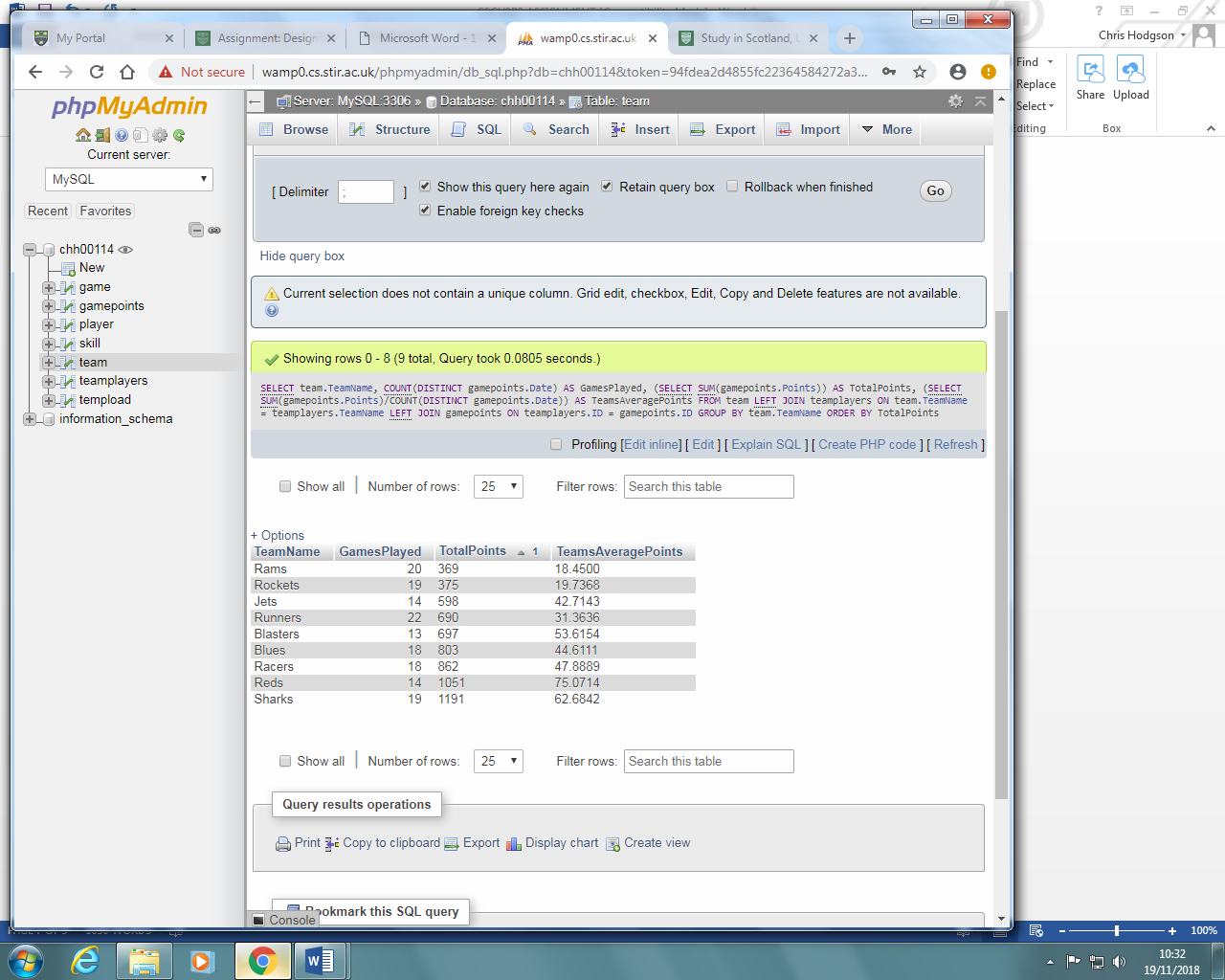
FROM team

LEFT JOIN teamplayers ON team.TeamName = teamplayers.TeamName

LEFT JOIN gamepoints ON teamplayers.ID = gamepoints.ID

GROUP BY team.TeamName

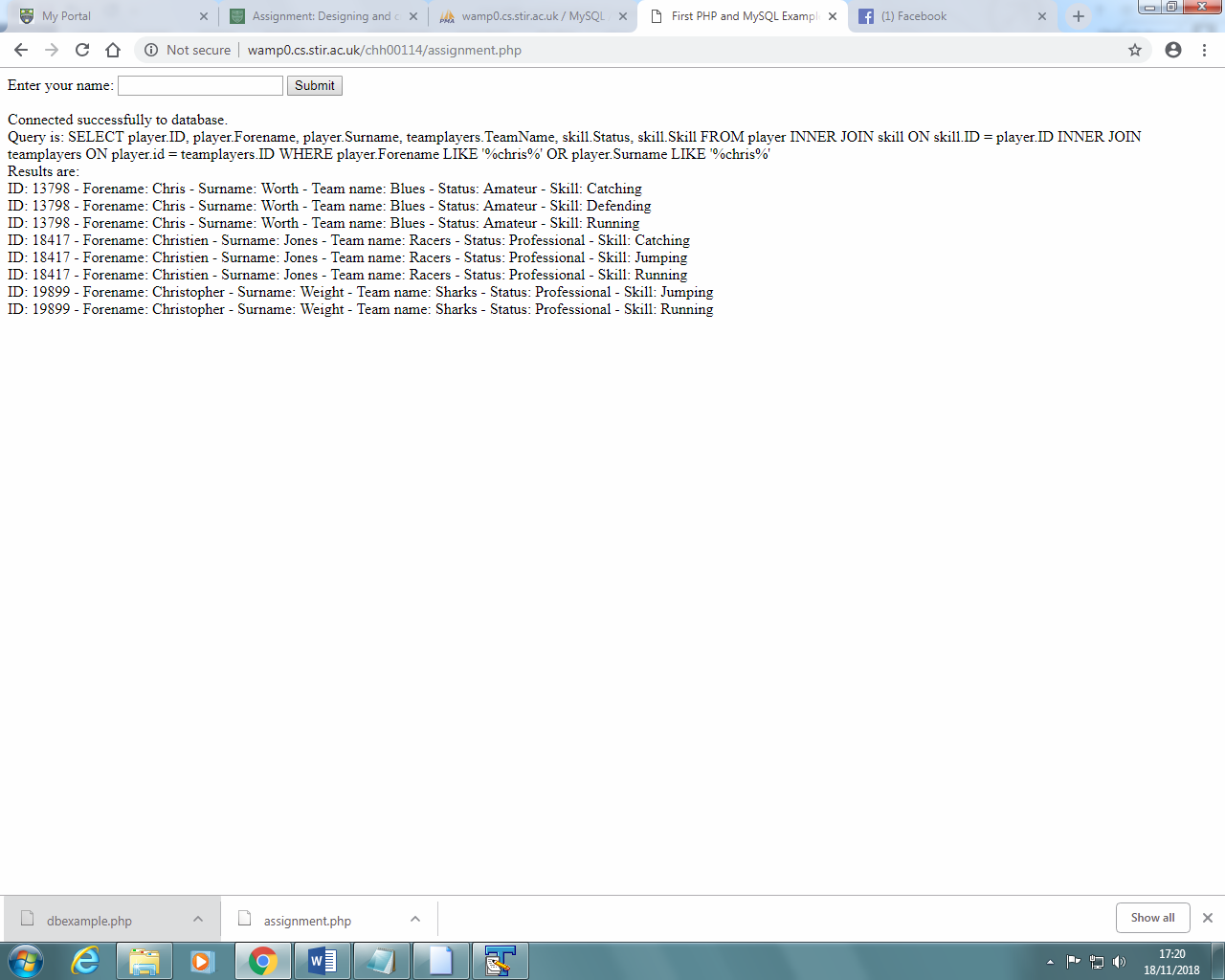
ORDER BY TotalPoints;



1. Starting with the template file, [**assignment.php**](https://canvas.stir.ac.uk/courses/3362/files/502543/download?verifier=7F1t8WK6q5bOIM0QXaLU67tZ6dcGJf053LbYI88t&wrap=1) (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:
2. Search for any players whose names (forename or surname) contain the text entered.
3. Display neatly in the web page the following characteristics of all matching players found: ID, forename, surname, team, status, skills
4. Try out your code by placing this file (**DO NOT rename it**) in your web folder on wamp0

**Link:** [**http://wamp0.cs.stir.ac.uk/chh00114/assignment.php**](http://wamp0.cs.stir.ac.uk/chh00114/assignment.php)

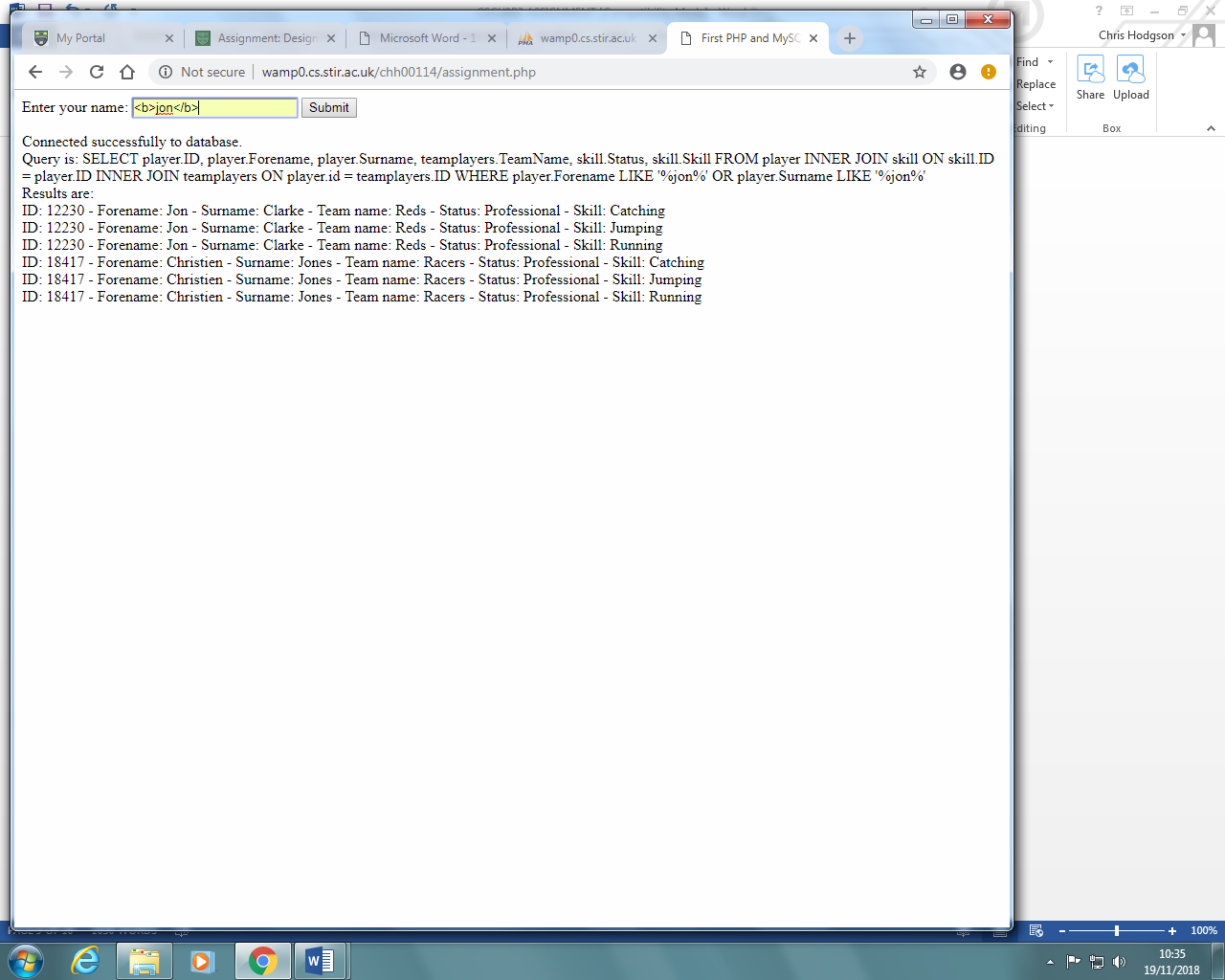
Search results for ‘chris’



Search results for ‘joyce’ which doesn’t exist.



Search results for “<b>jon</b>” testing tag stripping



1. **In your report**, include a copy of your code

<html>

<head>

<title>First PHP and MySQL Example</title>

</head>

<body>

<form action ="assignment.php" method="post">

Enter your name: <input type="text" name="name">

<input type="submit">

</form>

<?php

// procedural version

$servername = "wamp0.cs.stir.ac.uk";

$username = "chh00114";

$password = "password";

$database = "chh00114";

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

$name = $\_POST['name'];

// 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 player.ID, player.Forename, player.Surname, teamplayers.TeamName, skill.Status, skill.Skill

FROM player

INNER JOIN skill ON skill.ID = player.ID

INNER JOIN teamplayers ON player.id = teamplayers.ID

WHERE player.Forename LIKE '%".$name."%' OR player.Surname LIKE '%".$name."%'";

echo "Query is: ".$sql."<br>";

$result = mysqli\_query($conn, $sql);

if (!$result) {

echo "Search produced an error: ". mysqli\_error($conn);

}

else {

echo "Results are:<br>";

// output data of each row

while($row = mysqli\_fetch\_row($result)) {

echo "ID: " . $row[0]. " - Forename: " . $row[1]. " - Surname: " . $row[2]. " - Team name: " . $row[3]. " - Status: " . $row[4]. " - Skill: " . $row[5]."<br>";

}

//added in here

if($result->num\_rows < 1) {

echo "No results found";

}

//ended here

}

mysqli\_close($conn);

}

?>

</body>

</html>