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:

ID	Forename	Surname	Team	Status	skill	Name	Town	Venue	Date	Points
10162	Gillian	Botwright	Racers	Amateur	Catching	Racers	B' of Allan	Doune	03/01/12	4
10162	Gillian	Botwright	Racers	Amateur	Jumping	Racers	B' of Allan	Doune	03/01/12	4

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:
    a. Put the table name at the top of each table
    b. List the fields in each table
    c. Underline the primary key fields
    d. Put a * after the foreign key fields
    e. Mark the cardinality of each relationship at both ends of the connecting line
    f. 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.
    a. List each team name and the town they are based in.
    b. List the total number of games played by each team.
    c. 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).
    d. List the dates of all the games where the Jets and the Rams both played.
    e. 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:
    a. Search for any players whose names (forename or surname) contain the text entered.
    b. Display neatly in the web page the following characteristics of all matching players found: ID, forename, surname, team, status, skills
    c. 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).
    d. **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.