CSCU9B3 Relational Database Assignment Spring 2023

Computing Science and Maths, University of Stirling

70% of module grade

Due: Midnight Wednesday 19th April 2023

A local boatyard hires out cruising yachts on the Forth. The season is 20 weeks long. Each week is identified by number: Week 1, Week 2, etc. There are several yachts: each has a unique name (for example, *Mhairi*). The lengths of the yachts are recorded in feet. Groups of people make bookings of a boat for a week: each group has a “skipper” and the remainder of the group is the “crew”. There are at least two people in each group. Some people may be in several parties (in different weeks) and the skipper for a booking in one week may be a crew-member in another week. The firm keeps information on all the people in the party, including the skipper. The information kept on people consists of a unique “person-number”, their name, and their e-mail address. When booking, the group may request one or more standard “extras” such as Gin and Tonic or Hot Water Bottles. Each extra is identified by a unique string, e.g. “Gin + Tonic”.

The firm has been badly advised and keeps its data in a single table in First Normal Form (1NF). Because there is only one table, the firm has to use several records to hold all the information about one booking. For example, if a party requests two extras, all the other information about the booking has to be included twice.

The table was created with the following statements:

CREATE TABLE BoatBooking  
 (Week NUMBER(2),  
 BoatName VARCHAR2(20),  
 BoatLength NUMBER,  
 ExtraRequest VARCHAR2(20),  
 SkipperPno NUMBER,  
 SkipperName VARCHAR2(40),  
 SkipperAddress VARCHAR2(40),  
 CrewPno NUMBER,  
 CrewName VARCHAR2(40),  
 CrewAddress VARCHAR2(40)) ;

We see that each row of the table gives information about the week-number that the booking is for; the name of the boat; the boat’s length in feet; an extra requested by the party (or else the string “(no extras)”); the skipper’s person-number, name, and email address; and the person number, name and email address of one of the crew. It is clear that the table is in 1NF but not 2NF, and should be decomposed into a number of smaller tables.

**It is your job to turn this data into a relational database.**

**The data is stored in the boatbookingraw.csv file which has no header information, for loading into the database, which you can download from the module’s Canvas assignment page.**

**Present your results with comments in a written report.** Follow the steps overleaf. These instructions are detailed and following them properly should ensure you get good marks.

1. Design a set of tables for a relational database conforming to third normal form to store this data. **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. Highlight the primary key fields with keyword “PK”
   4. Highlight the foreign key fields with keyword “FK”
   5. Mark the cardinality of each relationship at both ends of the connecting line
   6. Indicate optionality with a dashed line or circle symbol.
2. **In your report**, write a justification for your design, considering aspects such as data integrity and normalisation. *Approximately 400 words.*
3. 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.
4. 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).
5. 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.
6. **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 use the tables from step 5; that is, make all your queries from the tables corresponding to your ER decomposition. (Hint: for part d you may find it convenient to use a view in constructing the query.)

For each query, explain the rationale behind your choice of query; that is, how you expect the particular SQL terms to help you retrieve the information required. For part d, comment on what your query tells you about the hypotheses. *Approximately 50 words per query explanation.*

* 1. Find the week-numbers and crew names for bookings of *Janet*.
  2. Display a table showing the total number of bookings for each week (i.e the number of boats booked in week 1, week 2, and so on).
  3. Find the name of the boats(s) with the largest number of extras requested over the season as a whole by creating a table of boat names and number of extras ordered in descending order by the number of extras. Do not count the ‘(no extras)’ items.
  4. Write queries to investigate the following two hypotheses:
     1. Shorter boats are booked more frequently.
     2. There is a relationship between number of extras, including no extras, and the size of the party.

1. 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 person whose names (forename or surname) contain the text entered.
   2. For each person found, check if that person is a skipper and which boats they skipper. For each person found display neatly their name and address and either “not a skipper” or boats skippered and week in which that boat is skippered. Use a sensible ordering for your query.
   3. Try out your code by placing this file (**DO NOT rename it**) in your C:\wamp64\www folder, as you did in practical exercises .
   4. **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 created for part 5.
   5. **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 upload it as .PHP file too.

# Marking breakdown and criteria

Marks will be awarded both for the technical correctness of what you have done and for the clarity and organisation of how you describe it.

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 and data transfer: 10%
4. Searches: 25%
5. PHP (Assignment.php): 15%
6. Overall quality of report: 10%

# Submitting your work and assessment procedures

**The assignment will be submitted as an electronic (PDF) type-written report uploaded (via Turnitin) to Canvas by the above-mentioned deadline. 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.

### Late submission

### If you cannot meet the assignment hand-in deadline and have good cause, please use the extension request link on the left-hand navigation to explain your situation and ask for an extension. Coursework will be accepted up to seven days after the hand-in deadline (or expiry of any agreed extension), but the mark will be lowered by three marks per day or part thereof. After seven days, the work will be deemed a non-submission.

### Plagiarism

### Work that is submitted for assessment must be your own work. Plagiarism means presenting the work of others as though it were your own. The University takes a very serious view of plagiarism, and the penalties can be severe (ranging from a reduced mark in the assessment, through a failure mark for the module, to expulsion from the University for more serious or repeated offences). We check submissions carefully for evidence of plagiarism and pursue those cases we find. Further details can be found here:

<https://www.stir.ac.uk/media/stirling/services/academic-registry/documents/Policy-and-Procedure-Academic-Integrity_v4_FINAL.docx>

### Note

### Students should be aware that a copy of their coursework will be retained in Canvas, and it may be used anonymously to create an exemplar answer for future students. If you do not wish your coursework to be used for this purpose, please inform the module co-ordinator upon submission using the Comments box.