



**UNIVERSITY  
OF LONDON**

**CM2040**

**BSc EXAMINATION**

**COMPUTER SCIENCE**

**Databases, Networks and the Web**

**Release date:** Tuesday 15 March 2022 by 12:00 midday Greenwich Mean Time

**Submission date:** Wednesday 16 March 2022 at 12:00 midday Greenwich Mean Time

**Time allowed:** 24 hours to submit

**INSTRUCTIONS TO CANDIDATES:**

**Section A** of this assessment paper consists of a set of **TEN** Multiple Choice Questions (MCQs) which you will take separately from this paper. You should attempt to answer **ALL** the questions in Section A. The maximum mark for Section A is **40**.

Section A will be completed online on the VLE. You may choose to access the MCQs at any time following the release of the paper, but once you have accessed the MCQs you must submit your answers before the deadline or within 4 hours of starting, whichever occurs first.

**Section B** of this assessment paper is an online assessment to be completed within the same 24-hour window as Section A. We anticipate that approximately **1 hour** is sufficient for you to answer Section B. Candidates must answer **TWO** out of the **THREE** questions in Section B. The maximum mark for Section B is **60**.

Calculators are not permitted in this examination. Credit will only be given if all workings are shown.

You should complete **Section B** of this paper and submit your answers as **one document**, if possible, in Microsoft Word or PDF to the appropriate area on the VLE. You are permitted to upload 30 documents. However, we advise you to upload as few documents as possible. Each file uploaded must be accompanied by a coversheet containing your **candidate number** written clearly at the top of the page before you upload your work. Do not write your name anywhere in your answers.

## **SECTION A**

Candidates should answer the **TEN** Multiple Choice Questions (MCQs) quiz, **Question 1** in Section A on the VLE.

## SECTION B

Candidates should answer any **TWO** questions in Section B.

### Question 2

In your new job as a backend developer for ship2europe.com - a shipping company - you are responsible for implementing a MySQL database system. Your task is to design and develop a web application allowing company workers around Europe to perform the following tasks:

- Create a **customer** profile given the following attributes: customer name, username, password, email and address.
- Create **shipments** defining cargo type, collection date, delivery date, pick-up point, destination, weight and volume
- Additional attributes are needed to update the shipment status based on how far through is the delivery process using a 0-5 scale i.e. 0 is at pickup point and 5 is at final destination. Also, in a text field to provide feedback regarding the delivery outcome e.g. cargo was damaged or missing etc.
- Create **distribution\_centres** handling the shipments attributed by freight storage capacity, arrival and departure dates of shipments.
- The customers can have several shipments and each shipment is destined to one distribution centre.

- (a) For the ship2europe database, create an Entity Relationship Diagram (ERD). Entities, relationships between entities, and appropriate association types should all be included in your diagram. Use only three tables in the diagram.

[9]

- (b) For each table of the database you designed in phase 1(a), list the primary and foreign keys.

[7]

- (c) Create **two** of the tables you mentioned in your answer to part 1(a) with the appropriate SQL code.

[14]

### Question 3

Consider the home insurance database below, where each line shows a table with its properties, TableName (column1, column2, ...), and the primary and foreign keys are underlined. First column in each table is a primary key and other underlined column names are foreign keys referring to the column with the same name in another table. As an example, 'case\_number' in 'Filled' table is a foreign key referring to primary key 'case\_number' in 'Incident' table.

*Customer* (customer\_id, name, cust\_address)

Customers table holding the name and current address

*Property* (property\_id, prop\_address)

Properties table holding the address

*Incident* (case\_number, year, type)

Incidents table holding the year of the incident and type e.g. flooding, burglary etc.

*Filled* (id, case\_number, property\_id, customer\_id, damage\_cost)

Filled table holding the damage cost and associated columns

- a) Describe what an SQL INNER JOIN statement is. Similarly define the RIGHT and LEFT JOIN operations. Give set theory visual examples of the 3 operations. [9]
- b) Write a SQL JOIN statement that finds total damage costs of incidents where the customer named 'Bill Crook' was involved. [7]
- c) Write a SQL statement to find all types of incidents where a customer named 'Bill Crook' was involved. [7]
- d) Write a SQL statement that finds the total number of incidents between 2010 and 2020 inclusive. [7]

#### Question 4

In your new role as a backend developer you are asked to use Express and Node.js, as well as a MySQL database management system, to create a web application for a 'shopping list'. Here is a list of tasks which users can have when running the application.

- Setup a user profile
  - Login using credentials
  - Add new 'item' to the list
  - Remove an 'item' from the list
  - List all items in the 'shopping list' sorted by price
- (a) Write some code for a route named 'create\_user' that provides for the following requirements and functionality:
- A template file named user.ejs is used to store form data
  - The user.ejs is already rendered via a different route of the web application
  - The *username* and *password* are part of the form data
  - Your code should access the database to store the user data

- Your code should handle an error condition where the database query fails
- Your code should display a message when a new user is successfully created

Please note that you are just required to write the middleware code, not the template file. Make sure that each section of the code is appropriately commented.

[10]

- (b) Write an EJS template file for creating a user account named user.ejs. Secondly, include a 'title' that is passed as a parameter to this EJS file.

[3]

- (c) Write a piece of code for a different route called 'login' to let a user login using the username and password already saved in the database as follows:

- To collect form data by a template named login.ejs similar to user.ejs you have already done in part 4(b)
- To compare collected form data to data already saved in the database
- The user can login if and only if both username and password match with data saved in the database
- Your code should handle an error condition where the database query fails
- When the username isn't discovered in the database or the password doesn't match, your code should display relevant error messages
- Remember that each username in the database is unique
- Finally, don't forget to include the necessary comments for each section of your code

[13]

- d) Write a template file (ejs) to display all items passed as a parameter from middleware to the template file. The template file should generate a complete HTML page, it should have a title using a suitable HTML tag, the incoming variable is called availableItems.

[4]

END OF PAPER