

BSc EXAMINATION**COMPUTER SCIENCE****Databases, Networks and the Web**

Release date: Tuesday 26 September 2023 at 12:00 midday British Summer Time

Close date: Wednesday 27 September 2023 by 12:00 midday British Summer Time

Time allowed: 4 hours to submit

INSTRUCTIONS TO CANDIDATES:

Part A of this assessment consists of a set of **TEN** Multiple Choice Questions (MCQs). You should attempt to answer **ALL** the questions in **Part A**. The maximum mark for Part A is **40**.

Candidates must answer **TWO** out of the **THREE** questions in **Part B**. The maximum mark for Part B is **60**.

Part A and Part B will be completed online together on the Inspira exam platform. You may choose to access either part first upon entering the test area but must complete both parts within **4 hours** of doing so.

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

Do not write your name anywhere in your answers.

PART A

Candidates should answer the **TEN** Multiple Choice Questions (MCQs) in Part A of the test area.

PART B

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

Question 1: Deliverator pizzas

You are developing a database for a new pizza delivery company called 'The Deliverators'. The pizza delivery company's main selling point is that customers must receive their pizza within 20 minutes or they will get their money back as well as keeping the pizza.

- The database contains information about different types of people: pizza chefs, pizza 'deliverators' (i.e delivery drivers) and customers. This is stored in a single table.
- The database contains information about pizza orders, including which pizzas are in the order, the customer they must be delivered to and the timing and status of the pizza order.
- Available types of pizza are stored in their own table. Pizza orders consist of one or more pizzas.
- Pizza orders have a status field which can be 'new', 'cooking', 'out for delivery' or 'delivered'.
- Pizza orders have the necessary time information to store how long the pizza spends in each status.

- a. Create a basic specification for the database. Name and briefly describe in English the purpose of the database tables you would create to model the data described above. Name the primary and foreign keys for each database table, explaining what the purpose is for each relation. You do not need to describe the detailed fields of the tables.

[10 marks]

- b. Create an Entity Relationship Diagram using crow's foot notation as seen in the course. This should show relationships between the tables you created in your answer to the previous part.

[3 marks]

- c. Write in SQL code and explain in English the queries to create the tables necessary to represent a pizza order. Make sure your data will allow the company to monitor the time taken to deliver pizzas. You do not need to define tables for users - just use foreign keys to point to user data. Specify suitable data types for the data in your tables.

[8 marks]

- d. A pizza deliverator called Hiro has managed to hack into the database. He wants to set all of his pizza delivery times to less than 20 minutes to avoid losing his job. Explain how he can go about doing this and write SQL queries for him.

[4 marks]

- e. Write in SQL code and explain in English a query that computes the average delivery time per deliverator (pizza delivery person). Your query must use JOINS. State which type of JOIN you are using and why. Explain why other forms of JOIN would not be appropriate.

[5 marks]

Question 2: Messaging applications

- a. Consider web-based email systems such as Gmail, Outlook, Hotmail etc. The low-level mail transport and storage happens using specialised email protocols and servers such as SMTP, IMAP and POP. The email is not stored in a traditional database. The web interface is a higher-level system permitting users to interact with those lower-level systems using a web browser. Do you think the three-tier web application structure is an effective way to model such a system? Consider each part of the three tier system and explain if it maps meaningfully to some part the email system described. Use a diagram to clarify your answer.

[6 marks]

- b. You are designing a three tier web application which allows its users to send text-based messages to each other. A user called Lauren wishes to send a message to a user called Adrian. Describe the journey of a message which starts with Lauren typing it and ends with Adrian reading it. Consider the three tiers - which tier is involved in each step? What computer languages, protocols and software are involved? Provide an example.

[6 marks]

- c. Write out a database schema for the messaging application. The system needs to store users and messages. For now, design the system so one message only goes to one user. Name the tables and explain how they link to each other using primary and foreign keys. List the fields in each database table and select appropriate data types.

[4 marks]

- d. You want to add a group messaging feature to your application. This allows users to create messages that should be delivered to more than one user. Explain how you will adapt the database schema to cope with this new feature.

[2 marks]

- e. Write an SQL query that will retrieve all messages for a given user from your group messaging-capable database.

[2 marks]

- f. You are working on the design for the routes of your web server. Considering the functionality and flow of data you outlined in the example above where Lauren sends a message to Adrian, name and describe the purpose of a set of routes suitable for your messaging application. Explain which database tables each route interacts with and what kind of SQL queries it will need to make. You do not need to write the SQL queries, but you do need to explain what kind of queries they are. Use a diagram if it clarifies your answer.

[6 marks]

- g. Explain how you would go about implementing security features for your messaging platform. Describe TWO security features.

[4 marks]

Question 3: Exam mark tracking system

You are developing a web application which allows university administrators to track exam marks. The database for the system has the following tables:

session : session_id (primary key), session_name

course : course_id (primary key), course_name

exam_mark : mark_id (primary key) session_id (foreign key). course_id (foreign key), student_id (foreign key), mark. status: not_marked, marked, marking_notes

When the students complete the exam, entries are created in the exam_mark table for each student who submitted the exam. Later, markers can use a web-based form to enter a mark for a student and to update the status from 'not_marked' to 'marked'.

- a. Create an EJS template which allows a marker to submit a student id, a mark and some marking notes for an exam. The template should present a suitable form to the user. The template receives data from the web application which is the marker name, course name and session name. The template should display this data and also embed it in hidden fields in the form. Here is some starter code for your template.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Mark entry form</title>
</head>
<body>
  <h1>Please enter your marks</h1>
  <form action="create-exam-mark-record" method="post">
    [1]
    <button type="submit">Create a new mark record</button>
  </form>
</body>
</html>
```

Please complete the template where the [1] appears.

[6 marks]

- b. Complete the following route such that it can receive the data from the exam marking form you created in the previous part.

```
router.post("/create-exam-mark-record",
  (req, res, next) => {
    // [1] prepare the data here for the SQL query
    const data = ...
    global.db.run(
      // put the appropriate arguments here
      // [2] arg 1: the SQL
      // [3] arg 2: the data for the query
      // [4] arg 3: the callback including handling of error
      condition and no-error condition
    );
  });
```

There are four items to complete in the route, labelled [1], [2], [3] and [4]. Please supply the code for each of these components.

Include full explanatory comments for each part of your code. You must explain each part of the code adequately. If you do not provide an explanation you will not receive any marks.

[8 marks]

- c. It turns out that the database server is sometimes unreliable. Sometimes you need to try a few times before the queries get through from your route code. Write some code that attempts to send SQL queries multiple times before giving up. Can you think of any problems with your implementation?

[2 marks]

- d. Write an ExpressJS route which displays a summary of how the exam marking is progressing for a given session. It should retrieve the data needed to list all the courses associated to a given `session_id` which is sent to the route as a parameter. Your route's job is to collect the necessary data from the database and to compose it in the following structure:

```
[
  { "course name": "CM2040 Databases, Networks and the Web",
    "exams submitted": 150,
    "exams marked": 10 }
  ... other courses from this session here
]
```

`exams submitted' is the total number of exams submitted for a course in a session.

`exams marked' is the total number of exams submitted for a course in a session where the status is set to `marked'.

You do not need to write a template for your route. You just need to write the code to gather the data and to structure it into the format shown above. You can use JOIN queries if you want or you can do multiple queries and compute the data in Javascript. It is up to you. Here is a framework for your route.

```
router.get("/get-marking-status[1]", (req, res, next) => {
  global.db.all() [2] complete this database query
  [3] add more queries if needed
  [4] compile the data into the needed format
  [5] send the data back to the requesting client
});
```

Please complete each of the missing parts [1] to [5].

Include full explanatory comments for each part of your code. You must explain each part of the code adequately. If you do not provide an explanation you will not receive any marks.

[10 marks]

- e. Identify a useful feature for tracking marks and explain how you would go about implementing it. Which templates and routes would be needed?

[4 marks]

END OF PAPER