



**BSc EXAMINATION**

**COMPUTER SCIENCE**

**Databases, Networks and the Web**

**Release date:** Wednesday 24 March 2021 at 12 midday Greenwich Mean Time

**Submission date:** Thursday 25 March 2021 by 12 midday Greenwich Mean Time

**Time allowed:** 24 hours to submit

**INSTRUCTIONS TO CANDIDATES:**

**Section A** of this assessment 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 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 a 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**. In addition, your answers must have 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 **ALL** of Question 1 in Section A.

### Question 1

- (a) Can we refer to any collections of data as 'databases' according to Silberschatz, Korth and Sudarshan authors of the famous database textbook? [4]

Choose ONE option:

- i. No, only collections of data that are highly valuable or relatively large and are accessed by multiple users and applications often at the same time are referred to as databases.
- ii. No, only collections of data that are highly valuable or relatively large and compressed are referred to as databases.
- iii. Yes, any collections of data are referred to as databases.
- iv. No, only collections of data that are highly valuable or relatively large and are accessed by multiple users and applications often at the same time and are accessed by Structured Query Language (SQL) are referred to as databases.

(b) Consider the employee database below, where each line shows a table with its properties, TableName (column1, column2, ...), and the primary keys and foreign keys are underlined. As an example, 'dept\_name' in the 'instructor' table is a foreign key referring to the primary key 'dept\_name' in the 'department' table. Which of the following SQL statements finds the name of the instructors teaching in the department with the lowest budget? [4]

*instructor (instructor\_id, name, dept\_name, salary)*  
*department (dept\_name, building, budget)*

Choose ONE option:

- i. SELECT instructor.name FROM instructor JOIN department WHERE department.budget ON instructor\_name is MIN;
- ii. SELECT name FROM instructor JOIN department ON instructor.dept\_name=department.dept\_name WHERE budget = (SELECT MIN(budget) FROM department);
- iii. SELECT name, MIN (budget) FROM instructor, department;
- iv. SELECT department.name FROM instructor JOIN department ON department.dept\_name = instrutor.name WHERE budget = (SELECT MIN(budget) FROM department);
- v. SELECT dept\_name FROM department WHERE budget = (SELECT MAX (budget) FROM department);

(c) Which of the following are **not** HTTP request methods?

Select ALL options that apply. [4]

- i. PATCH
- ii. SEND
- iii. GET
- iv. PUT
- v. LIST

**(d)** Which layer of the TCP/IP model does the three-hand shake process relate to? [4]

Choose ONE option:

- i. Link layer
- ii. Application layer
- iii. Transport layer
- iv. Internet layer
- v. Network layer

**(e)** Which of the following terms suit the following definition 'a constantly-running piece of software listening to requests and providing responses accordingly to users using HTTP'?

Select ALL statements that apply. [4]

- i. A Node.js file including 'listen' function of the 'express' module
- ii. A webserver software
- iii. TCP/IP software suite
- iv. A Node.js file including 'createServer' function of the 'http' module
- v. Database Management System (DBMS)
- vi. HTML software standard

**(f)** Which one of the following correctly corresponds to the three tiers of the three-tier web application architecture? [4]

Choose ONE option:

- i. Front-end, middleware, and business logic tiers
- ii. Presentation, user interface, and business logic tiers
- iii. The user interface, middleware, and business logic tiers
- iv. Presentation, business logic, and data tiers

**(g)** Which of the following are valid SQL statements to change a record in a table named 'Jewellery'? The 'Jewellery' table has three properties of id, name and price of types INT, DECIMAL and VARCHAR respectively.

Select ALL statements that apply.

[4]

- i. CHANGE Jewellery VALUES (NULL, 'Ring', 30.25);
- ii. UPDATE Jewellery SET price=30.25 WHERE name='Ring';
- iii. UPDATE Jewellery SET (name, price) VALUES ('Ring', 30.25);
- iv. UPDATE Jewellery SET price=30.25;
- v. UPDATE Jewellery VALUES ('Ring', 30.25);

**(h)** Which of the following best list benefits of the 'SoC (Separation of Concern)' principle of programming?

Select ALL statements that apply.

[4]

- i. Improves testability and portability, increases the program's run-time speed
- ii. Reduces the complexity of the program, improves maintainability, portability and testability
- iii. Improves portability, increases the complexity of the program, and helps the DRY (Don't Repeat Yourself) principle of programming
- iv. Increases the program's run-time speed, improves testability, and helps the DRY (Don't Repeat Yourself) principle of programming
- v. Improves portability, reduces the complexity of the program, and helps the DRY (Don't Repeat Yourself) principle of programming

(i) Which of the following are TRUE about the order of the code execution in the sample main.js file below?

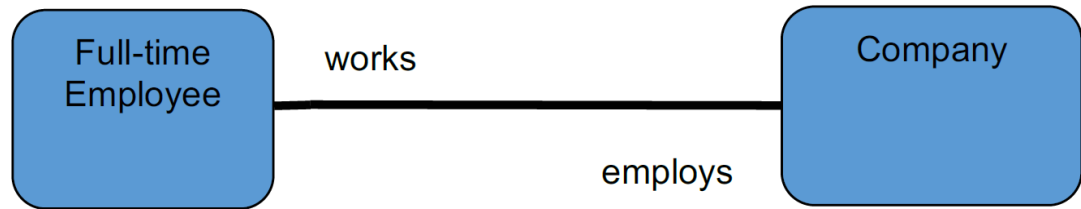
```
1. app.get('/about',function(req,res){
2.     res.render('about.html');
3. });
4. app.get('/search',function(req,res){
5.     console.log('The search keyword is:'+
6.     req.query['keyword']);
7.     res.render("search.html");
8. });
```

Select ALL statements that apply.

[4]

- i. This is an event-driven programming style and line number 4 is always the first line to be executed.
- ii. In event-driven programming, each line of code is executed after the other as coded by the programmer.
- iii. Line number 4 is only executed in response to an event related to this function.
- iv. Line number 1 is only executed when the corresponding route is typed in the URL because this is not an event-driven programming style
- v. Line number 5 is always executed after line number 4.
- vi. Line number 4 (search function) is always executed after line number 3.

- (j) Which of the following statements best completes the below Entity-Relationship (ER) Diagram considering the relationship between 'Full-time Employee' and 'Company' entities? [4]



Choose ONE option:

- i. There is a zero-to-many relationship between 'Full-time Employee' and 'Company' entities
- ii. There is a one-to-one relationship between 'Company' and 'Full-time Employee' entities
- iii. There is a one-to-many relationship between 'Full-time Employee' and 'Company' entities
- iv. There is a many-to-many relationship between 'Company' and 'Fulltime Employee' entities
- v. There is a one-to-many relationship between 'Company' and 'Full-time Employee' entities

## SECTION B

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

### Question 2

Consider the university database below, where each line shows a table with its properties, *TableName (column1, column2, ...)*, and the primary and foreign keys are underlined. The 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, 'dept\_name' in the 'instructor' table is a foreign key referring to the primary key 'dept\_name' in the 'department' table. Please note each instructor teaches only one department and each department has only one building. Each building may belong to more than one department and each course is taught only at one department but by one or more instructors.

*instructor* (instructor\_id, name, dept\_name, salary)

*department* (dept\_name, building\_id, budget)

*building* (building\_id, building\_name, location)

*classroom* (classroom\_id, building\_id, room\_number, capacity)

*course* (course\_id, title, dept\_name, credits)

You are a freelance backend developer responsible for implementing a relational database for a university, for which you were given the information above.

- (a) Imagine we had a list of comma and tab delimited data instead of the university database above, explain by an example, what would be the 'delete' modification issue associated with lists of data compared to the university database above. Illustrate with a diagram if necessary. [7]
- (b) List all foreign keys in each table of the above university database. [4]
- (c) Draw an Entity Relationship diagram (ERD) for the university database. Your diagram should include entities and relationships between entities and suitable association types. [10]
- (d) Do you see any problem in the above ERD? Propose a solution to resolve the problem and to consider the teaching scenario in a university. Write the corresponding SQL code if your solution includes adding a new table to the above design. [9]



### Question 3

Consider the university database below, where each line shows a table with its properties, *TableName (column1, column2, ...)*, and the primary and foreign keys are underlined. The first column in each table is a primary key and the other underlined column names are foreign keys referring to the column with the same name in another table. As an example, dept\_name' in the 'instructor' table is a foreign key referring to the primary key 'dept\_name' in the 'department' table.

*instructor (instructor\_id, name, dept\_name, salary)*

*department (dept\_name, building\_id, budget)*

*building (building\_id, building\_name, location)*

*classroom (classroom\_id, building\_id, room\_number, capacity)*

*course (course\_id, title, dept\_name, credits)*

- (a) Present an example of a SQL nested SELECT or SELECT in SELECT statement for the above database and then explain what it does and where it should be used. [5]
- (b) Write a SQL statement that finds the total number of classrooms in the building with the id 21. [4]
- (c) Write a SQL statement that finds the capacity of the room number 303 in the building named 'White building'. [6]
- (d) Write a SQL JOIN statement that lists the name and salary of the instructors in the 'computing' department with the lowest salary of all the departments. [7]
- (e) Write a SQL statement to list the names and salaries of all the instructors in the 'computing' department who earn more than the minimum salary in the department. [8]

#### Question 4

You are a full-stack developer building a dynamic web application for diabetes patients to input and store their daily food and insulin intake and exercise activities in a database. The application lists users' input on a daily, weekly and monthly basis to help them keep their blood sugar level in balance. You have written the web server code named 'index.js' as shown below:

```
1. var express = require ('express')
2. var bodyParser= require ('body-parser')
3. var mysql = require ('mysql')
4. const app = express()
5. const port = 8000
6. const db = mysql.createConnection ({
7. host: 'localhost',
8. user: 'root',
9. password: 'password',
10. database: 'BalancedSugar'
11. });
12. db.connect((err) => {
13. if (err) {
14. throw err;
15. } console.log('Connected to database');
16. });
17. global.db = db;
18. app.use(bodyParser.urlencoded({ extended: true }))
19. require('./routes/main')(app);
20. app.set('views', __dirname + '/views');
21. app.set('view engine', 'ejs');
22. app.engine('html', require('ejs').renderFile);
23. app.listen(port, () => console.log(` Example app listening on port ${port}!`))
```

- (a) What are the steps you would follow as a full-stack developer for the complete design of a relational SQL database for this dynamic web application? [6]
- (b) What pages (routes) would you suggest to be designed for the implementation of this web application? [8]
- (c) What is the name of the database created for this web application based on the index.js code you can see above? [2]
- (d) What is the message displayed when the web server starts to run based on the index.js code you can see above? [2]
- (e) Explain the purpose of importing 'body-parser' module on line 2. [2]

(f) Write a piece of middleware code, with a route named 'add\_activity' as follows:

- To store form data collected by a template file named activity.ejs.
- Form data include the type of the activity (such as walking, running, cycling, etc), the number of minutes spent on the activity and the date.
- 'add\_activity' saves those data in the database.
- Your code should access the database to store the activity.
- Each activity stored in the database has three properties: 1) the type of activity, 2) the minutes spent on the activity and 3) the date.
- Your code should handle the error condition where the database query fails.

Please note, you are only required to write the middleware code and not the template file. Make sure you add comments for each section of the code. [7]

END OF PAPER