UNIVERSITY OF LONDON

CM3010

BSc EXAMINATION

COMPUTER SCIENCE

Databases and Advanced Data Techniques

Release date: Tuesday 8 March 2022 at 12:00 midday Greenwich Mean Time

Submission date: Wednesday 9 March 2022 by 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 a PDF to the appropriate area on the VLE. 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 the **TEN** Multiple Choice Questions (MCQs) quiz, **Question 1** in Section A on the VLE.

SECTION B

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

Question 2

A historian has developed an XML file to keep track of the English monarchy of the 16th Century. An extract from the file is reproduced below.

```
<royal name="Henry" xml:id="HenryVII">
  <title rank="king" territory="England" regnal="VII"
         from="1485-08-22" to="1509-04-21" />
  <relationship type="marriage" spouse="#ElizabethOfYork">
    <children>
      <royal name="Arthur" xml:id="ArthurTudor"/>
      <royal name="Henry" xml:id="HenryVIII>
        <title rank="king" territory="England" regnal="VIII"
               from="1509-04-22" to="1547-01-28" />
        <relationship type="marriage" spouse="#CatherineOfAragon"
                      from="1509-06-11" to="1533-05-23">
          <children>
            <royal name="Mary">
              <title rank="queen" territory="England" regnal="I"
                     from="1553-07-19" to="1558-11-17" />
              <relationship type="marriage" spouse="#PhilipOfSpain"
                            from="1554-07-25"/>
            </royal>
          </children>
        </relationship>
        <relationship type="marriage"  spouse="#AnneBoleyn"
                      from="1533-01-25" to="1536-05-17" />
          <children>
            <royal name="Elizabeth">
              <title rank="queen" territory="England" regnal="I"
                     from="1558-11-17" to="1603-03-24" />
            </royal>
          </children>
        </relationship>
        <relationship type="marriage" spouse="#JaneSeymour"
                      from="1536-05-30" to="1537-10-24">
          <children>
```

```
      <royal name="Edward">
        <title rank="king" territory="England" regnal="VI"
               from="1547-01-28" to="1553-07-06" />
      </royal>
    </children>
  </relationship>
  ...
```

(a) Give two examples of element names and two examples of attribute names from this code. [2]

(b) What will be the result of the following XPath query: `//title[@rank="king" and @regnal="VIII"]/../royal[@name="Henry"]`? [3]

(c) What (in general terms) will be returned by the following XPath query: `//title[@rank="king" or @rank="queen"]/../relationship/children/royal/relationship/children/royal/` ? [3]

(d) Mary I of England was also queen consort of Spain from 16 January 1556 until her death. Give an XML fragment that would record this information and say where you would add it to the code above. [4]

(e) The historian argues with colleagues about the strengths and weaknesses of this approach, using XML, and this model in particular. What are the strengths and weaknesses? [7]

(f) One colleague suggests that the data is really a graph not a tree, so should be represented as **Linked Data** using RDF. The other thinks it can be modelled as a set of relations and so should be transformed into a **relational database**. Who is correct? [1]

(g) Choosing one of the two suggested approaches (relational database or RDF), explain (with examples) how it might solve the strengths and weaknesses you listed in (e) above. [10]

Question 3

Wikidata provide a SPARQL query interface to their user-contributed, encyclopedia data.

The following Wikidata URIs may help you answer elements of the question below:

- `wdt:P19`: "place of birth"
- `wdt:P31`: "instance of" (wikidata's version of `rdf:type`)
- `wdt:P131`: "located in the administrative territorial entity" (an entity's geographical location)
- `wdt:P734`: "Family name"
- `wdt:P735`: "Given name"
- `wd:Q5`: "human"
- `wd:Q60`: "New York City"

(a) What will the following query return? [2]

```
SELECT DISTINCT ?person
  ?person wdt:P31 wd:Q5;
          wdt:P19 wd:Q60.
```

(b) What assumptions does this query make? What data must be present for it to work? [2]

(c) How does the following query differ? Does it resolve any of the assumptions you listed in (b) above? [4]

```
SELECT DISTINCT ?person
  ?person wdt:P31 wd:Q5;
          wdt:P19/wdt:P131* wd:Q60.
```

(d) The results of these queries are not particularly human readable. Why not? [1]

(e) How would you rewrite the query given in (c) to return something more readable? [5]

(f) The Internet Movie DataBase (IMDB) provides a way to retrieve actors based on their place of birth (for example, the following URL returns a list of movie-related people born in Queens in New York, `https://www.imdb.com/search/name/?birth_place=Queens,%20New%20York%20City,%20New%20York,%20USA`). This search is not exposed by IMDB in their web API for searching programmatically. Compare the IMDB approach and the Wikidata approach. [6]

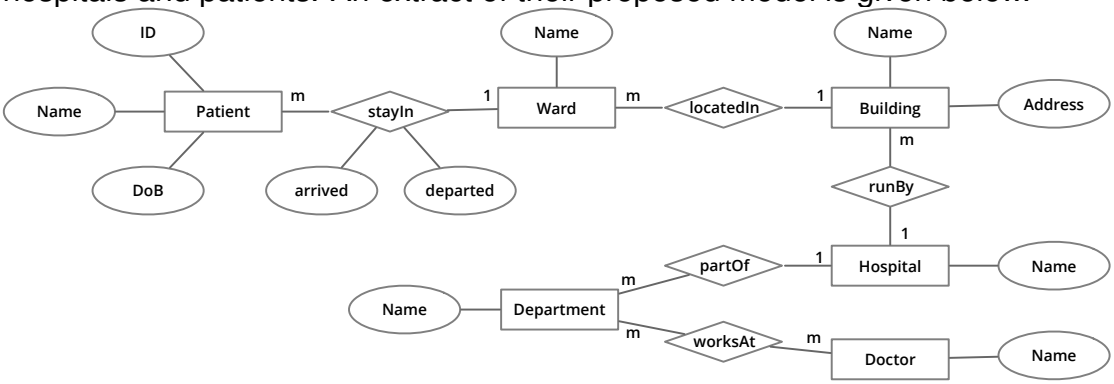
(g) IMDB has a lot of specialised information about movies that may not be available in Wikidata. How might you combine the strengths of the Wikidata and IMDB? [4]

(h) How would you represent the information queried in (b) using the relational model? Illustrate your model with a comparable query in SQL. [2]

(i) How would you approach the version in (c) in SQL? [4]

Question 4

A health organisation is designing a database to keep track of its doctors, hospitals and patients. An extract of their proposed model is given below.



A patient stays in a ward (which holds many patients), and the model records the arrival and departure dates. Each ward is one of many in a building, and individual hospitals will run at least one building. Hospitals are organised into departments, such as Orthopedics, Accident and Emergency, or Ear, Nose and Throat, and a doctor can work in multiple departments in multiple hospitals.

(a) Which of the following questions could be answered by an implementation of this model? [3]

i. Which building did the patient named Neha Ahuja stay in?

ii. Which hospital was responsible for Neha Ahuja's stay?

iii. In which wards are Orthopedics patients housed?

iv. Which hospitals does the doctor Song Ci work in?

v. What departments does the hospital have that contains a building called 'The Alexander Fleming Building'?

vi. Which doctor treated Neha Ahuja?

(b) Part of this model cannot be implemented using the relational model. Which part, and how would you resolve it? [3]

(c) Adapt the model so that all questions in part (a) and the issue you identified in (b) are resolved. Include cardinality in your diagram. [10]

(d) List the tables and keys for an SQL implementation of your model (you do not need to list fields here). [5]

(e) For each of the questions in (a), provide an appropriate MYSQL query. [6]

(f) Would this data structure work better using a tree based model such as XML? Give reasons for your answer. [3]

END OF PAPER

.