

## PART A

Candidates should answer **ALL** of Question 1 in Part A.

### Question 1

- (a) Normalisation can be thought of as an attempt to reduce information duplication in a database. What is the **ONE** primary reason why this is desirable? (note: several of these reasons are valid)

Choose ONE option

[4]

- i. Database storage was expensive when the theory was formalised. Even a reduction of 10% on storage requirements could save thousands of pounds.
- ii. Duplicate information can get out of sync, leading to logical inconsistencies in the data
- iii. Duplicate information takes more time to input. Adding extra structure through normalisation speeds data entry.
- iv. Redundant data represents a significant security threat, since it provides patterns in your database that can be exploited by hackers

- (b) I want to query a database table of books and join it to a table of authors. Where the book is anonymous, I want the author field to have the value NULL. What type of join do I need?

Select ONE correct option

[4]

- i. LEFT JOIN
- ii. INNER JOIN
- iii. CROSS JOIN
- iv. INNER JOIN and special WHERE clause

- (c) Look at the following sequence of commands and then choose the statement from the list that should replace XXXXXXXX

```
START TRANSACTION;  
SELECT stockLevel FROM Stock WHERE ID=23948267;  
UPDATE Stock SET stockLevel=stockLevel-2 WHERE ID=23948267;  
INSERT INTO Orders (productID, quantity) VALUES (23948267, 2) WHERE ID=292484  
XXXXXXX
```

Choose ONE option.

[4]

- i. GRANT ALL ON \* TO 'Stock' WITH GRANT OPTION;
- ii. UPDATE Stock SET stockLevel=2;
- iii. END TRANSACTION;
- iv. COMMIT;
- v. UPDATE Stock SET stockLevel=stockLevel+2 WHERE ID=23948267;

(d) What does the following MongoDB call do?

```
db.books.find({
  title: /*man/,
  author: /*man/,
  year: 1934
});
```

Choose ONE option.

[4]

- i. Sets the year to 1934 for books with a title or author called man
- ii. Finds all books with title and author called man
- iii. Finds all books with title and author ending in man
- iv. Finds all books with title and author called man with a year of 1934
- v. Finds all books with title and author ending in man with a year of 1934
- vi. Adds a new record to books with title "man", author "man" and year 1934

(e) Why are URLs important in Linked Data and other Semantic Web technologies?

Select ALL correct statements.

[4]

- i. A URL is necessarily unique, and so can be used like a Primary Key
- ii. A URL is shareable, so anyone can refer to it when they want to show they mean the same thing
- iii. A URL can encode meaning in its path that can be parsed into useful information
- iv. HTTP requests for a URL can be 'dereferenced' – they can have responses that provide useful information about the thing the URL represents

- v. A URL is permanent and reliable, providing important guarantees for any data
- (f) Given the XML fragment below, what does the XPath expression `//note/title` refer to?

```
<msContents>
  <textLang mainLang="en">English</textLang>
  <msItem n="1" xml:id="Christ_Church_MS_473-item1">
    <locus> fol. 1{26</locus>
    <rubric>Certain humble Petitions which are in most humble manner...
    </rubric>
    <incipit>1. That there may be a veiwe taken of all the
              markett Townes</incipit>
    <note>Printed by John Strype, <title>Annals of the Reformation</title>
          (London, 1709), 3:Appendix (68{81), working from what is now BL,
          MS Lansdowne 119. Strype, <title>Annals</title>, 3:221 attributes
          both the Petition and the related 'Supplication', which follows in
          our codex, to the Puritan divine,
          <persName key="person_228371281"
                    role="aut">Thomas Sampson (d. 1589)</persName>.
    </note>
  </msItem>
  <msItem n="2" xml:id="Christ_Church_MS_473-item2">
    <locus> fol. 28{34<hi rend="superscript">v</hi></locus>
    <author key="person_228371281"/>
    <incipit defective="true">and painfull preachinge pastors
                              resident among us</incipit>
    <explicit>obey you accordinge to his blessed will thorowly
    </explicit>
    <note>The complete text, titled <title>A Supplication made
          in the Name of certain true subjects...</title> and opening
          'In most humble wise complaining, we which are Thousands
          of the poor untaught People of England...', is printed by Strype,
          <title>Annals</title>, 3:222{27. Our copy has lost the title
          and first 30 lines, as printed by Strype, equating presumably
          to one recto and verso. </note>
  </msItem>
</msContents>
```

Select ONE correct statement.

[4]

- i. The four `title` elements that appear as direct child nodes of `note` elements
- ii. Nothing, since no child of the root element is a `note` elements
- iii. The first `title` element that in a `note`
- iv. A `title` element, but there are none in this fragment
- v. The two `note` elements that have at least one child element that is a `title`

(g) Which of the following are true statements about MapReduce?

- i. The Map phase is carried out on local data
- ii. The Reduce phase is carried out on local data
- iii. The Map phase produces data that can be distributed based on key
- iv. The Reduce phase produces data that can be distributed based on key
- v. The input data for MapReduce systems can be distributed
- vi. Reducer operations can be distributed
- vii. MapReduce reduces the amount of processing needed
- viii. MapReduce makes it easier to run the processing in parallel

Select ALL correct statements.

[4]

(h) Consider the following code:

From the music ontology:

```
mo:listened a owl:ObjectProperty
    dfs:comment "Relates agents to the
                performances they were listening in" ;
    rdfs:domain foaf:Agent ;
    rdfs:range mo:Performance ;
    owl:inverseOf mo:listener .
```

And from some published data:

```
orcid:0000-0003-4151-0499 a foaf:Agent ;
    mo:listened <http://data.carnegiehall.org/events/46018> .

<http://data.carnegiehall.org/events/46018>
```

dct:date 2008-03-28T19:30:00 ^^xsd:dateTime .

Given these statements, which of the following **MUST** be true?

- i. orcid:0000-0003-4151-0499 a mo:Performance .
- ii. <http://data.carnegiehall.org/events/46018> a foaf:Agent .
- iii. <http://data.carnegiehall.org/events/46018> a mo:Performance .
- iv. orcid:0000-0003-4151-0499 dct:date "2008-03-28T19:30:00"8sd:dateTime .
- v. <http://data.carnegiehall.org/events/46018> mo:listener orcid:0000-0003-4151-0499 .

Select ALL correct statements.

[4]

- (i) A retrieval algorithm has a fairly constant 20% precision across all result sets. If my corpus contains 15,030,482 documents of which 225,030 are relevant, which of the following statements are likely to be true?
- i. A result set of 80 documents will have about 32 correct matches.
  - ii. A result set of 80 documents will have about 16 correct matches.
  - iii. There will be at most about 45,000 correct matches returned.
  - iv. The algorithm does over 10 times better than random.
  - v. The algorithm does over 100 times better than random.

Select ALL correct statements.

[4]

- (j) Which of the following statements is true of **Copyleft**
- i. Copyleft licences are described as **permissive**
  - ii. Copyleft licences are not **permissive**
  - iii. Copyleft licences allow derivative works freedom to choose their own licensing systems
  - iv. Copyleft licences restrict the licensing of derivative works to ensure that they are also Copyleft
  - v. The GNU Public Licence (GPL) is an example of Copyleft
  - vi. The MIT licence is an example of Copyleft

Select ALL correct statements.

[4]

## PART B

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

### Question 2

The following text describes a television series:

Doctor Who was first broadcast in 1963, with William Hartnell playing what we now call the 'First Doctor'. From the start, the Doctor was accompanied by companions. The first companions were his granddaughter, Susan Foreman, and two of her schoolteachers, Ian Chesterton and Barbara Wright. Susan was played by Carole Ann Ford.

In 1966, an episode was broadcast in which the Doctor collapses from exhaustion. His body regenerates into the 'Second Doctor', played by Patrick Troughton.

In total, between 1963 and the present, there have been 13 numbered incarnations of the Doctor, and one extra incarnation called 'The War Doctor', along with dozens of companions. Although most companions appear exclusively with one Doctor, some overlap with several, and some take guest appearances in episodes with later Doctors.

- (a) We wish to design a simple database about this series, modelling Doctors, their companions, who played them and when they were broadcast. List the tables needed, indicating all keys. [6]
- (b) Give a MySQL command for creating ONE of these tables [3]
- (c) Are your tables in 2NF? How can you tell? [3]
- (d) Give MySQL commands to answer the following queries in your schema:
  - i. Who played the Doctor whose companion was Amy Pond? [2]
  - ii. Was (companion) Peri featured before Leela? (or was Peri's Doctor before Leela's?) [4]
  - iii. Which incarnation of the Doctor had the most companions? [3]
- (e) Here is an extract from the DBpedia entry for the First Doctor:

```
dbr:First_Doctor rdfs:label      "First Doctor"@en ;  
                  dbp:periodEnd  "1966-10-29"^^xsd:date ;
```

```

dbp:periodStart "1963-11-23"^^xsd:date ;
dbp:companions  "Ben Jackson"@en ,
                "Vicki"@en ,
                "Sara Kingdom"@en ,
                "Steven Taylor"@en ,
                "Susan Foreman"@en ,
                "Polly"@en ,
                "Ian Chesterton"@en ,
                "Barbara Wright"@en ,
                "Dodo Chaplet"@en ,
                "Katarina"@en ;
dct:subject     dbc:Doctor_Who_Doctors ;
dbp:next        dbr:Second_Doctor .

```

- i. What serialization language is this? [1]
- ii. How many triples are encoded here? [1]
- iii. What can your database schema do that this approach can't? [2]
- iv. How would you fix that problem? [2]
- v. Fix the following SPARQL query, which should look for any incarnation of the Doctor who had Amy Pond as a companion

```

SELECT dbr:doctor
WHERE {
    dbr:First_Doctor dbp:next+ ?doctor .
    dbr:doctor dbp:companion "Leela" .
}

```

[3]

### Question 3

(a) Consider the following XML fragment

```
<castList xmlns="http://www.tei-c.org/ns/1.0">
  <castGroup>
    <castGroup>
      <head>four lovers</head>
      <castItem xml:id="Hermia_MND">
        <role>
          <name>Hermia</name>
        </role>
      </castItem>
      <castItem xml:id="Lysander_MND">
        <role>
          <name>Lysander</name>
        </role>
      </castItem>
      <castItem xml:id="Helena_MND">
        <role>
          <name>Helena</name>
        </role>
      </castItem>
      <castItem xml:id="Demetrius_MND">
        <role>
          <name>Demetrius</name>
        </role>
      </castItem>
    </castGroup>
  </castGroup>
  <castGroup>
    <castItem xml:id="Theseus_MND">
      <role>
        <name>Theseus</name>
      </role>
      <roleDesc>duke of Athens</roleDesc>
    </castItem>
  </castGroup>
```

- i. This fragment is unbalanced. What is missing
- ii. What format is this?

[1]

[1]



iii. What attributes are used in this fragment?

[2]

(b) Here is an edited extract from the associated xsd file

```
<xs:element name="castGroup">
  <xs:annotation>
    <xs:documentation>(cast list grouping) groups one or more individual cast.
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:group ref="ns1:model.global"/>
        <xs:group ref="ns1:model.headLike"/>
      </xs:choice>
      <xs:sequence maxOccurs="unbounded">
        <xs:choice>
          <xs:element ref="ns1:castItem"/>
          <xs:element ref="ns1:castGroup"/>
          <xs:element ref="ns1:roleDesc"/>
        </xs:choice>
      </xs:sequence>
      <xs:attributeGroup ref="ns1:att.global.attributes"/>
    </xs:complexType>
  </xs:element>
```

i. What is this file and what does it do?

[2]

ii. The castGroup element as defined here begins with model.global or model.headlike elements. Neither happens the first time this element occurs in the fragment above. Does this mean the document is invalid? Explain your answer.

[3]

iii. Do the castGroup elements in the fragment follow this definition correctly? Explain your answer.

[3]

(c) Consider this extract of an xsl file associated with this fragment:

```
<xsl:template match="castItem">
  <div>
    <xsl:apply-templates match="node()|@*"/>
  </div>
</xsl:template>
<xsl:template match="role/name">
```

```

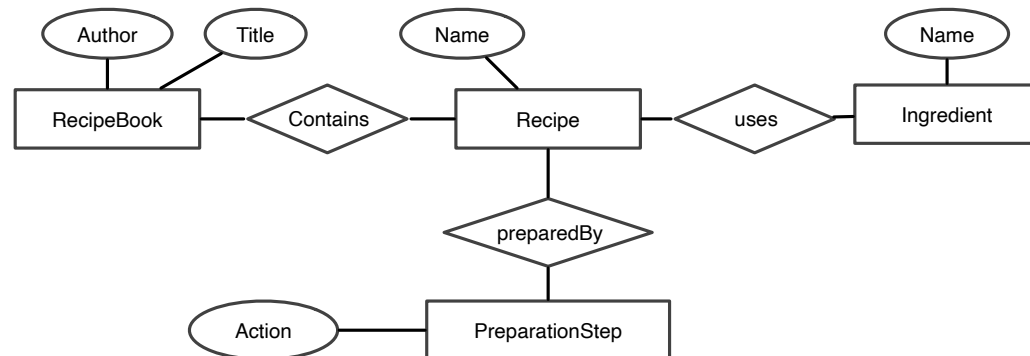
<dt>
  <xsl:value-of select="."/>
</dt>
</xsl:template>
<xsl:template match="roleDesc">
  <dd>
    <xsl:value-of select="."/>
  </dd>
</xsl:template>

```

- i. What is this file and what is it for? [2]
  - ii. This fragment won't work. What is missing from the `match` attributes? [2]
  - iii. What will be the format and content of the output of applying this file (when it works)? [3]
- (d) You have been tasked with representing the same data, but using the relational model
- i. What tables would you use to represent this cast information? [6]
  - ii. Which works better here, the relational model or the XML implementation? Why? [5]

#### Question 4

A friend has sketched their idea for a recipe database in an E/R diagram. This database will power a website that will allow browsing by ingredient, or by recipe books and their authors, as well as just by recipe name.



- Do you think there's anything missing from this model that the database might need? If so, what? [4]
- What is the cardinality of the three relationships? [3]
- Give MySQL-compatible CREATE TABLE commands for TWO of the tables that you would use to implement the database (including any corrections from (a) above). Include any appropriate key constraints. [10]
- Give a single MySQL query to find all the recipe books which never use butter. [4]
- Give a single MySQL query to find the average number of steps in the recipe book called "Mushrooms" by "John Cage". [4]
- Consider ONE of the following technologies and indicate whether you think they would be more suitable for this database. Explain your answer. [5]
  - Document database (MongoDB)
  - XML document resource or database
  - Linked Data resource or triplestore

END OF PAPER