University of London

Assessment Coversheet

Complete this coversheet and read the instructions below carefully.

Candidate Number: EX0829
Refer to your Admission Notice

Degree Title: BSc
e.g. LLB

Course/Module Title: Databases and Advanced Data Techniques
As it appears on the question paper

Course/Module Code: CM3010
This is in the top right corner of the question paper. If there is more than one code, use the first code.

Enter the numbers, and sub-sections, of the questions in the order in which you have attempted them:
Q2 (a) (b) (c) (d) (e) (f) (g) (h) (i)

Q4 (a) (b) (c) (d) (e)

Date: 06 MARCH 2023

Instructions to Candidates

1. Complete this coversheet and begin typing your answers on the page below, or, submit the coversheet with your handwritten answers (where handwritten answers are permitted or required as part of your online timed assessment).
2. Clearly state the question number, and any sub-sections, at the beginning of each answer and also note them in the space provided above.
3. For typed answers, use a plain font such as Arial or Calibri and font size 11 or larger.
4. Where permission has been given in advance, handwritten answers (including diagrams or mathematical formulae) must be done on light coloured paper using blue or black ink.
5. Reference your diagrams in your typed answers. Label diagrams clearly.

The Examiners will attach great importance to legibility, accuracy and clarity of expression.

Begin your answers on this page

Q2 Candidate Number: EX0829

a) This code snippet is encoded in XML (Extensible Markup Language), which is commonly used for creating structured documents or data formats

(b) It uses hierarchical structure, with each element having a unique name and potentially containing attributes or child elements. For example, "office:text" is the root element while others, such as "text:p" "text:list" and "text:list-item" served as child elements.

(c) The two namespaces are:

① The default namespace, as the "office" prefix used in the "office:text" element, is used since the document is using the Open Document Format (ODF) standard.

② The text namespace, which is associated with the "text" prefix, is used for elements that define the documents content.

(d) The first expression "//text:list-item/text:p", this Xpath expression returns all "text:p" elements that are direct children of "text:list-item" elements in this document.

The second expression "//text:list //text:p" would return all the "text:p" elements that are descendants of any "text:list" element in the XML document, regard less of how depth within the tree.

So the difference that the first expression only selects those "text:p" elements that are direct children of "text:list-item" elements. but the second expression returns all "text:p" elements that are anywhere within "text:list" element.

Q2 (e) Candidat Number: EX0829

The provided code snippet defines the structure and content requirements for OpenDocument files. Specifically, it derfines the rules for the "text:list" and "text:list-header" elements. For example, the code rules that a "text:list" element must contain the "text-list-attr" element, and may optionally contain a "text-list-header" element and zero or more "texe-list-item" elements. If the document does not fulfill these requirements, it would be considered as not well-formed.

(f) The code defines the structure and content requirements for OpenDocument files, including the attributes, header (optional), and list items (zero or more). By comparing the document to this schema, if it has an element that is not allowed by the schema, or if it is missing an element that is required by the schema, the document would be considered as invalid.

(g) It is relevant to the "text:list" element, its attributes, header and list items.

Q2

(h) The example would be like:

```
<text:list>
  <text:list-header>
    <text:p> List Header </text:p>
  </text:list-header>
  <text:list-item>
    <text:p> Item 1 </text:p>
  </text:list-item>
  <text:list-item>
    <text:p> Item 2 </text:p>
  </text:list-item>
</text:list>
```

Above example, this "text:list" element is missing the "text-list-attr" element, which is required by the schema code, and only contains the optional "text-list-header" and "text-list-item" elements. So it would be considered as invalid according to the schema.

(i) This data structure is suitable for encoding word processing documents to a certain extent. Because it includes elements such as "text:p" for paragraphs, "text:list" for lists and more, which are commonly used in word processing documents.

Advantage of Relational Model:

It allows for creating more complex data structure that can include multiple tables, and can be linked together through relationships, which allows for a greater flexibility and scalability when dealing with larger or more complex documents. Besides, It has more efficient data management, as it can be optimized for performance and handle larger datasets more effectively.

Disadvantage:

The Relational model may be more complex to implement and maintain, which requires more advanced technique skills and It may also be less compatible and interoperable with some office applications.

Q4 Candidate Number: Ex0829

(a) One possible solution could be adding metadata to associate each line of music with a unique identifier or label. This label could be stored alongside the data for each line and used to sort the lines in correct order when they are retrieved.

(b) To achieve this, we could modify the existing data model to include two new entities: Part and Region.

The "Part" entity refers to the instruments or voice parts that make up pieces. Each Part could associated with a Piece, and could have attributes such as, part number, name, etc.

The "Region" entity could includes the different regions or sections of a page, each with lines going in different directions. Each Region would be linked to a Page, and could have attributes such as region number, etc.

Additionally, we can add a foreign key on the existing "Line" entity to link with the "Part" entity so that we can know which part each line of music belongs to

Candidate Number: EX0829

Q4(c) Tables: (* stands for primary key)

- Library (LibID*, Name, Location)
- Book (BookID*, shelfmark, Title, Number, composer, LibID)
- Page (PageID*, Number, BookID)
- Line (LineID*, LNumber, Xcord, Ycord, PID, RID)
- Part (PID*, PName, BookID)
- Region (RID*, XCord, Ycord, PID)
- Piece (PieceID*, Title, composer)

Primary keys:

Name	Table
LibID	Library
BookID	Book
PageID	Page
LineID	Line
PID	Part
RID	Region
PieceID	Piece

Foreign keys:

- BookID (Book) refers to BookID (Page)
- LibID (Book) refers to LibID (Library)
- BookID (Part) refers to BookID (Book)
- PageID (Region) refers to PageID (page)
- PID (Line) refers to PID (Part)
- RID (Line) refers to RID (Region)

(d)

```
SELECT p.title, COUNT(L.LineID) as total_lines
    FROM Piece p
    JOIN Line l ON p.PieceID = L.LineID
    GROUP BY p.PieceID
```

Q4 (e) Candidate Number: EX0829

The modified model could be well-suited for a relational model since the entities and relationships are clearly defined and the primary and foreign keys have been appropriately identified. It also allows efficient querying and manipulation of the data, as well as the ability to enforce data integrity constraints.

Compared to the XML-based tree database, the relational model has a more rigid schema and enforces data integrity. Since data is organized into tables with pre-defined relationships between them. it must conform to the relationships.

However, XML-based tree is more flexible since it was designed to store data as a hierarchical tree structure, making it suitable for storing and querying data with complex nested relationships which can be useful in certain situations. as it can handle more diverse data structures.