BCS THE CHARTERED INSTITUTE FOR IT

BCS HIGHER EDUCATION QUALIFICATIONS
BCS Level 6 Professional Graduate Diploma in IT

ADVANCED DATABASE MANGEMENT SYSTEMS

Monday 25th September 2017 - Afternoon
Answer **any** THREE questions out of FIVE. All questions carry equal marks.
Time: THREE hours

Answer any Section A questions you attempt in Answer Book A
Answer any Section B questions you attempt in Answer Book B

The marks given in brackets are **indicative** of the weight given to each part of the question.

Calculators are **NOT** allowed in this examination.

Section A
Answer Section A questions in Answer Book A

A1

a) The following ‘data models’ are alternatives to the Relational data model: -

Object oriented
Document oriented
Geographical or spatial

(i) Briefly explain the meaning of the term ‘data model’ given the above context. **(3 marks)**

(ii) With the aid of examples, explain how each of the above data models differ from the relational model.

(12 marks)

b) Recently there has been a lot of interest in alternatives to relational database systems known colloquially as 'NoSQL' databases and often referred to as 'Not Only SQL'. These alternatives have emerged to handle the massive growth of data - so called Big Data. Examples of Big Data applications include Facebook (and social media in general) and Google Analytics (Intelligent Product marketing).

With reference to the above statement, discuss the main requirements and challenges of developing and running Big Data applications.

(10 marks)

A2

a) Explain the distinction between ‘Semi-structured data’ as exhibited by data description languages such as XML and JSON and ‘Structured data’ as exhibited by the relational model.

(4 marks)

b) Outline the advantages of using semi-structured data over structured data typified by the relational model of data.

(5 marks)

c) Consider the XML document and XSL stylesheet code listings in Figures A2.1 and A2.2 below.

(i) Show the result of running mystylesheet.xsl on borrowers.xml. **(5 marks)**

(ii) Write an XPath expression that returns all the names of borrowers, who have at least one loan. **(3 marks)**

d) Refer to the XQuery expression in Figure A2.3 and borrowers.xml in Figure A2.1 below.

(i) Describe the function of the code. **(4 marks)**

(ii) Compare and contrast XQuery and XPath **(4 marks)**

Figure A2.1: XML document borrowers.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet href="mystylesheet.xsl" type="text/xsl"?>
<loans>
<borrower>
   <name>
   <first>Michael</first>
   <last>Hansen</last>
   </name>
   <address>Freisha Way, 23456 Copenhagen, Denmark</address>
   <loan>
        <amount>1000</amount>
        <payout-date>2016-01-01</payout-date>
        <repayment amount="100" date="2017-01-01"/>
        <repayment amount="100" date="2018-01-01"/>
   </loan>
   <loan>
        <amount>500</amount>
        <payout-date>2017-01-01</payout-date>
        <repayment amount="100" date="2015-01-01"/>
   </loan>
</borrower>
<borrower>
   <name>
        <first>Mansha</first>
        <last>Daleep</last>
   </name>
   <address>Costoria Street 8232, 98764 Mumbai, India</address>
</borrower>
</loans>
```

Figure A2.2 XSL stylesheet, mystylesheet.xsl

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="loans">
   <html>
   <body>
        <xsl:apply-templates select="//loan"/>
   </body>
   </html>
</xsl:template>
<xsl:template match="loan">
<xsl:apply-templates select="../name/last"/>,
<xsl:apply-templates select="../name/first"/>:
<xsl:apply-templates select="amount"/><br/>
</xsl:template>
</xsl:stylesheet>
```

Figure A2.3 Query expression

```
<borrowers>
{
for $x in doc("borrowers.xml")//borrower
return <borrower>
{$x/name}
<debt> {fn:sum($x/loan/amount) - fn:sum($x/loan/repayment/@amount)}
</debt>
</borrower>
}
</borrowers>
```

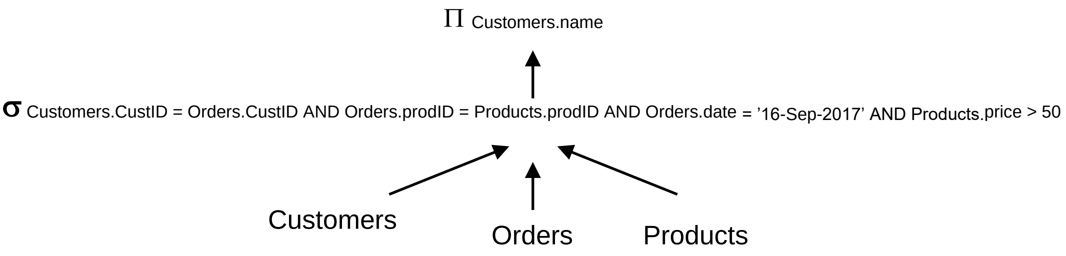
A3

a) Given the following three linked tables:

```
Customers (custID, name, country)
Products (prodID, price)
Orders (orderID, custID*, prodID*, date)
```

Suppose we have the following query and its corresponding initial parse tree:

```
SELECT Customers.name
FROM Customers, Orders, Products
WHERE Customers.custID = Orders.custID
AND    Orders.prodID = Products.prodID
AND      Orders.date = ’16-Sep-2017’
AND      Products.price > 50;
```


Π Customers.name
σ Customers.CustID = Orders.CustID AND Orders.prodID = Products.prodID AND Orders.date = ’16-Sep-2017’ AND Products.price > 50
Customers
Orders
Products

(i) What problems arise if the query is executed based on the above parse tree?

(2 marks)

(ii) Transform the above parse tree into one that corresponds to the most efficient way of processing the query.

(11 marks)

(iii) Which indexes could be created to potentially further enhance the execution of this query?

(2 marks)

b) A company employee has been using the password "APPLE" for the past six months to access a database. Discuss why this poses a security risk and suggest ways in which the company could improve password management.

(10 marks)

Section B
Answer Section B questions in Answer Book B

B4

Using your own simple examples and suitable diagrams, explain the following data warehousing and big data concepts. **Five marks each.**

a) Data Cleansing

(5 marks)

b) Indexing & Optimization

(5 marks)

c) Aggregation & Summarized Data

(5 marks)

d) Materialized Views

(5 marks)

e) Roll-up, drill-down and slicing & dicing

(5 marks)

B5

Using your own simple examples and suitable diagrams, explain the following transaction-processing concepts. **Five marks each.**

a) Two-phase locking

(5 marks)

b) Two-phase commit

(5 marks)

c) Cascaded rollback

(5 marks)

d) Wait-for graphs

(5 marks)

e) Checkpoints

(5 marks)