**BCS THE CHARTERED INSTITUTE FOR IT**

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

# ADVANCED DATABASE MANAGEMENT SYSTEMS

**Friday 23rd March 2018 - 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

## Section A
## Answer Section A questions in Answer Book A

**A1**

### GENERAL COMMENTS

This was a very popular question, attempted by the vast majority of candidates, with 70% achieving a pass mark.

This question relates to concurrency control in the context of a multi-user on-line transaction processing environment.

a) Explain, using examples from a database application (such as student registrations on courses), how the so-called **A.C.I.D.** properties of database transactions assist database integrity.

**(8 marks)**

### ANSWER POINTERS

Examples from a student record/registration system

**Atomic:** ensures each transaction is executed completely or not at all

Example:
Either a student has registered for course or they haven't. The state of registration must be known. Partial registration makes no sense and might leave the DB in an inconsistent state if the system crashes for example half way through an update.

**Consistent**: Database consistency is maintained from the start of a transaction until the transaction commits or aborted.
For example a transaction that registers a new student must consider a constraint that limits the number of students registered on a course to a particular value. The value cannot exceed the limit set in the DB. The transaction must maintain a consistent state irrespective of the activation of the constraint.

**Isolated:** The concurrent execution of a set of transactions has the same effect as an equivalent serial execution of that set. Isolation is therefore concerned with data integrity when there are sets of possible interacting transactions.
For example two transactions may concurrently register a student on a particular course. Both transactions read the current total of attendances is 29 and check the size limit (set at 30) for a particular course. One transaction commits and will register a student successfully and without isolation. The second transaction would also receive a value of 29 and hence be allowed to proceed commit the registration. This violates database integrity as the number of attendances will exceed the size limit.

**Durable:** The result or effects of committed transactions are permanently recorded in the database.
Once a student has been registered for a course the system must be durable enough to maintain the registration even if the system crashes. This involves writing the effects of the transaction to stable storage and ensuring transactions are logged and are recoverable - otherwise database integrity may be lost.

b) Explain, with aid of examples, the difference between *serial* and *serialisable* schedules of transactions. Comment on whether (and if so, how) one is a superset of the other.

**(5 marks)**

**ANSWER POINTERS**

A diagram is useful showing the example sequences of the read/write operations for two transactions in the order they are executed

Serial Schedule - no interleaving

- Transactions execute fully.
- One at a time. (hence, no interleaving).
- Different orders of execution may produce different final values

Serializable Schedule - interleaved

- Equivalent to SOME serial schedule.
- Equivalence does NOT mean "ending up with the same values as".
- Equivalence cannot depend on initial values of database items.
- Cannot depend on values written TM doesn't know logic of transaction.
- Depends only on order of operations.

Comments on whether a serializable schedule is a superset of a serial schedule.
Depends on whether schedules are equivalent and do not conflict (act the same as a serial schedule)

Example

Transaction A and Transaction B each Read and Write to resource X

Conflicting Operations:

- Trans A Read X and Trans B Write X
- Trans A Write X and Trans B Read X
- Trans A Write X and Trans B Write X
- Trans A Read X and Trans B Read X do not conflict

c) Briefly explain how database transactions can be :-

(i) blocked.

(ii) deadlocked.

**(3 marks)**

**ANSWER POINTERS**

Transactions can be blocked when a lock is applied to a shared resource (e.g. a table) and the transaction has to wait until the lock is released when the resource is free to read/write to it. Deadlocking occurs when there is a sequence of transactions in which say transactionA is waiting to access a shared resource held by transactionB and transactionB is waiting to access the same shared resource as TransactionA.

d) Describe a range of techniques that can be used to counter or minimise the impact of deadlocks in a highly concurrent system. Comment on the effectiveness of each technique.

**(5 marks)**

**ANSWER POINTERS**

Popular practical techniques for example:

- Minimise the possibility by designing short duration transactions.
- Counter by choosing one transaction involved in a chain of transactions in deadlock and loop

back the chain on itself rolling back each transaction in sequence. Some analysis needs to be done as to which one is chosen. Normally it would be the one that has done the least amount of work. Causes loss of work.

- Transactions are executed in some predefined order. This could be applicable to a series of overnight batch processes. Not really useful in OnlineTransactionProcessing (OLTP)

- Timeouts detect transactions that have been waiting a long time and assume a deadlock exists and the transaction is aborted. Problem here is that it is hard to automate and relies on human intervention.

- Timestamping - ordering of transaction execution is decided before transaction starts. May cause difficulty in starting transactions.

Coverage of at least four techniques at some depth of understanding will gain full marks

e) Briefly describe row level locking and table level locking and comment on the pros and cons of applying these types of locks.

**(4 marks)**

**ANSWER POINTERS**

In the case of row level locking only a small amount of data is locked compared to table level locking, thereby achieving higher levels of concurrency and hence performance in terms of increases in transaction throughput. The downside is that row level locking is susceptible to phantom reads following an insert. To ensure consistency in reading a table would require a lock on all rows in the table. However, this would not prevent new rows being inserted during the execution of the transaction, giving rise to inconsistencies. If a table level lock were to be applied no rows could be inserted.

**EXAMINERS' COMMENTS**

A number of candidates produced near correct solutions. The main weaknesses were identified in parts a), c) as follows:
Part a) at this level familiarity with the ACID acronym was expected and was mainly achieved. However, applying examples to each term was poorly executed in a lot of cases and hence marks were lost as a result.
Part c) A number of candidates demonstrated little understanding of blocking and some found it hard to distinguish blocked transactions from deadlocked transactions. It was expected that candidates should cover at least four of the techniques mentioned in the answer pointer. Timestamping was the most popular with slight variations on the other techniques generally accepted.

**A2**

**GENERAL COMMENTS**

This was the least popular question in Section A; It was attempted by approximately a fifth of all candidates. Although the least well answered question, over half of the attempts achieved pass level marks. The performance of candidates is quite varied - reflected in the wide range of marks with some candidates producing near perfect solutions.

.

This question relates to XML and/or related technology.

a) Outline the main differences between the following data models:-

(i) Document Oriented.

(ii) Relational.

**(4 marks)**

**ANSWER POINTERS**

A document-oriented database stores, queries, and maintains document-oriented information. This type of information is also known as semi-structured data. Document-oriented databases can be described in different ways, for example, key-value store in the JSON description language or in XML which is the key part to this question and listed on the syllabus. Data in document databases contrast strongly with the traditional relational database (RDB). Relational databases are strongly typed during database creation, and store repeated data in separate tables that are defined by the programmer.
SQL is the usual way to define an RDB in the form of a schema. A script of SQL code precisely defines the table's structure including constraints.

b) Explain why the use of XML and/or any related technology is ideally suited to achieve the following objectives:-

(i) Represent document-oriented data (instead of using a relational database).

(ii) A means of data exchange over the WWW.

**(8 marks)**

c) What is an XSL stylesheet? Show the result of applying the XSL stylesheet in Fig A2 to the XML document in Fig A1. Give a brief explanation of how the result is obtained.

**(5 marks)**

Fig A1 students.xml document:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet href="mystylesheet.xsl" type="text/xsl"?>
<studentloan>
 <student>
  <name>
   <first>Nicky</first>
   <last>Melson</last>
 </name>
 <address>Forest Way 1, NR236HF, Burgh, Norfolk</address>
 <loan>
  <amount>5000</amount>
  <paymentdue>2020-01-01</paymentdue>
  <repayment amount="120" date="2017-05-01"/>
  <repayment amount="100" date="2017-06-02"/>
 </loan>
 <loan>
  <amount>9500</amount>
  <paymentdue>2021-01-01</paymentdue>
  <repayment amount="400" date="2017-01-01"/>
 </loan>
 </student>
 <student>
  <name>
   <first>Majeev</first>
   <last>Khan</last>
```

```
  </name>
  <address>Circus Street 8, YO438GT, Malton, N. Yorkshire</address>
  </student>
</studentloan>
```

Fig A2 xsl stylesheet

```
<?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="studentloans">
<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>
```

**ANSWER POINTERS**

Relational database systems cannot meet all the demands of electronic business because they process data independently of its context. Traditional databases may be well suited for data that fits into rows and columns, but cannot adequately handle data presented in rich data structures such as nested data structures or complex documents, which are characteristic of typical Web content.
In contrast document databases get their type information from the data itself, they normally store all related information together, and allow every instance of data to be different from any other. This makes them more flexible in dealing with change and optional values, maps more easily into program objects, and often reduces database size. This makes them attractive for programming modern web applications, which are subject to continual change.

XML originally came into common use for the purpose of providing a common framework for the interchange of data over the Internet. XML standards provide a set of rules for the construction of Internet Media Types for use when sending XML. It also defines the types "application/xml" and "text/xml", which inform only that the data is in XML, and nothing about its semantics.

XML: provides a universal data format for integrated electronic business solutions and is ideally suited to document oriented data that is referred to as semi-structured data. In other words, the restriction of holding data in tables and the rules of normalisation are relaxed. An XML document is stored in a human readable text file.

*The unique strengths of using XML as a software data format for data exchange over the WWW include:*

*Simple syntax*
*Easy to generate and parse.*
*Support for nesting*
*Tags easily allow programs to represent structures with nested elements.*
*Easy to debug*
*Human-readable data format is easy to explore and create with a basic text editor.*
*Language and platform independent*

*XML and Unicode guarantee that your datafile will be portable across virtually every popular computer architecture and language combination in use today.*
*The benefits of the above mean that that is relatively easy to exchange of data between User A and User B over the WWW because the format of a document that is required by User B is formatted and transmitted using a common standards and protocols. Therefore, User B will receive the document and can easily check that they have been supplied with a valid document.*

This is crucial when passing data between separate systems, where a deviation from the expected format might mean that the data cannot be processed (or worse, is processed incorrectly).

It is important to address Namespace support - the ability to mix data intended to be read by multiple sources (or written by multiple sources) in the same document.

An example of this in action is the SOAP protocol - namespaces allow for the separation of the SOAP "Envelope", or "Wrapper" data which is passed alongside the serialised application data. This allows web frameworks to process and handle the SOAP Envelope and then pass the body / payload data onto the application.

d) Write an XPath expression that returns all the names of students who have had at least one loan.

**(3 marks)**

**ANSWER POINTERS**

XPath will navigate through the XML document for each loan element, where it exists, for each student. The code to do this would look like:-

```
//student[loan]/name
```

Explanation

// Selects nodes in the document from the current node that match the selection criteria
/ Selects from the appropriate node
[loan] Is a predicate that checks if a loan exists for that student. Predicates are embedded in square brackets and are used to find a node that contains a specific value.

Therefore, the query will return the one occurrence the full name of the student that has a loan element :

```
NickyMelson
```

e) Describe the function of the XQuery code listed in Fig A3 and determine the output produced when it is run against the XML document in Fig A1.

Fig A3 XQuery code listing:

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

```
</studentloan>
```

**(5 marks)**

## ANSWER POINTERS

For each `student` in `studentloan`, xml computes the total amount (the sum of the numbers in the amount elements), minus the sum of the numbers in the repayment attribute of the repayment elements. Outputs are in XML, each student name and outstanding amount in a new debt element.

```
<studentloan>
<student>
<Nicky><Melson>
<debt> 13800 </debt>
</student>
</studentloan>
```

## EXAMINERS' COMMENTS

This question related to XML and/or related technology and has appeared regularly on previous question papers. The first two parts are descriptive questions. The first part covers alternative database models to relational databases to support the emerging technologies that support 'big data'. It is pleasing to see that candidates are showing a growing awareness of the alternatives to traditional relational databases. Future candidates are advised to read the latest articles on 'big data' and data analytics and the new database trends that underpin the technology.
The second part is specific to XML and associated techniques. This was generally answered well and has become a stock question to draw out the differences between relational and document oriented databases.

Part c) requires a derivation of XML code. Many candidates did not demonstrate awareness that the XSL file acts as a query language that extracts specific parts of the document following the navigation paths through the XML document structure. This restricts access to data in the document looking for conditions that match. Navigation starts from the root node and returns the last name and first name where they exist and where there is a loan amount. Only one student in the students table meets this criteria. This returns as HTML markup.

Parts d) & e) Most candidates who attempted these parts achieved the correct solutions.

**A3**

## GENERAL COMMENT

Nearly all candidates attempted this question, and most attempts achieved pass level marks.

a) Denormalization is often used as a way of tuning the performance of a database.

(i) Discuss the advantages and disadvantages of denormalization.

**(4 marks)**

(ii) Using an example of a database design and query, show how denormalization can be used.

**(4 marks)**

## ANSWER POINTERS

(i) (1 mark each)

Advantages include:

- Speeds up data retrievals by reducing joins

Disadvantages include:

- Introduces redundancy which may result in inconsistencies
- Makes implementation more complex
- Slows down updates

(ii) The student can provide an example of design involving joins between tables, and where denormalisation could mean duplicating some of the attributes in order to speed up a query. For example, given the tables:

Student (StudentNbr, StudentName)
Module (ModuleNbr, ModuleTitle)
Results (StudentNbr, ModuleNbr, Grade)

And the query:

```
SELECT  StudentNbr, StudentName, ModuleNbr, Grade
FROM Results, Student
WHERE Results.StudentNbr = Student.StudentNbr
```

The attribute `StudentName` could be copied into `Results`, so that the query will only read from the `Results` table without the need for a join with the `Student` table. (4 marks)

**EXAMINERS' COMMENTS**:

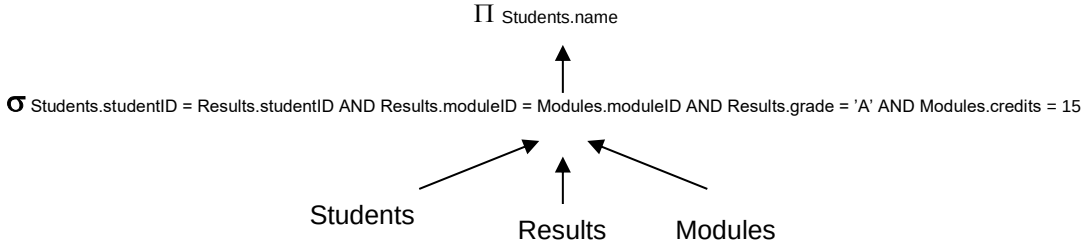
Many candidates managed to identify the benefit of denormalisation in reducing joins in queries but failed to provide a relevant example.

b) Given the following three linked tables:

```
Students (studentID, name, course)
Modules (moduleID, title, credits)
Results (resultID, studentID*, moduleID*, grade)
```

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

```
SELECT Students.name
FROM Students, Modules, Results
WHERE Students.studentID = Results.studentID
AND Results.moduleID = Modules.moduleID
AND Results.grade = 'A'
AND Modules.credits = 15;
```

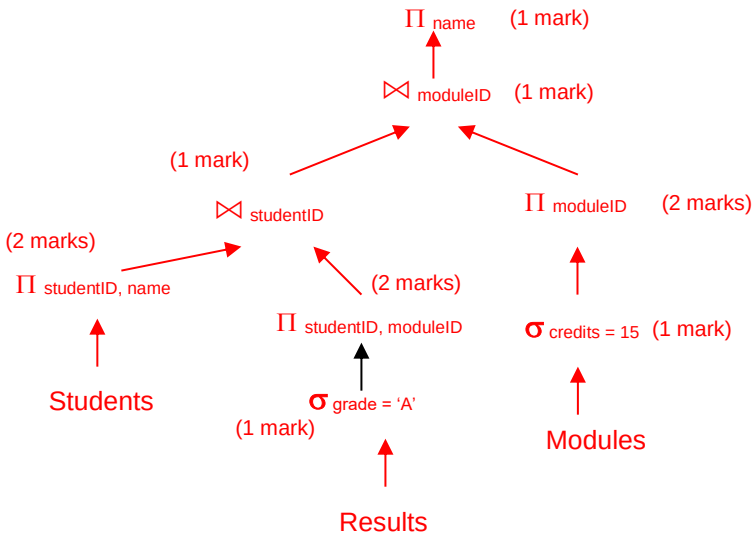



Transform the above parse tree into one that corresponds to the most efficient way of processing the query.

**(11 marks)**

## ANSWER POINTERS

A more efficient parse tree could be:




## EXAMINERS' COMMENTS

Almost all candidates managed to produce an optimised query tree.

c) Describe an appropriate database security control for each of the following security threats:

- Ransomware.
- SQL injection.
- An employee leaving the company.
- Brute-force password attacks.
- An employee making mistakes while entering data.
- A hacker intercepting data sent over the internet.

**(6 marks)**

## ANSWER POINTERS

Credit given for the following points:

- Ransomware: making regular backups of the database and testing those backups.
- SQL injection: encapsulating SQL code in stored parameterized procedures where data is sanitized.
- An employee leaving the company: preventing access by revoking privileges
- Brute-force password attacks: adopting a robust password policy
- An employee making mistakes while entering data: enforcing integrity constraints
- A hacker intercepting data sent over the internet: encrypt sensitive data

## EXAMINERS' COMMENTS

There were some good answers in general. Candidates need to focus on solutions related to the database as requested in the question. For example, although using an Antivirus is a sensible way of defending against ransomware, database backups are still needed.

## Section B
## Answer Section B questions in Answer Book B

**B4**

### GENERAL COMMENTS

This question was answered by less than half of the candidates.

You are a **database consultant** bidding for a new contract with a prestigious blue-chip organization. Part of the selection process is a technical interview. Answer the following questions posed by the interview panel.

a) With reference to a sample relation of your own choosing, explain and discuss the following relational model terminology, including its function and any related concepts.

- Tuple.
- Attribute.
- Domain.
- Constraint.

*A diagram showing your sample relation is strongly suggested.*

**(10 Marks)**

b) Using your own SQL examples, illustrate the various types of joins that may be employed when extracting data from one or more database tables.

**(10 Marks)**

c) Within the realm of distributed databases, explain the terms 'replication' and 'fragmentation'.

**(5 Marks)**

### ANSWER POINTERS

To be marked holistically but as a guide, 0-3 marks per item, up to a maximum of 10 marks overall. Full marks only for covering all the following points with clear examples. Tuple means row. Each row is uniquely identified by a primary key. Attribute means column. Each column must have a unique name within that table (relation) and can be isolated across tables by qualifying the column name with the table name (such as `student.name`). The domain specifies what are acceptable data values within a column (and by implication the acceptable operations on that column's data). Constraints can include primary keys, foreign keys, NOT NULL, CHECK and UNIQUE. Good diagram gets bonus marks.

### EXAMINERS' COMMENTS

For those who attempted this question, it was clear that they were very comfortable with the relational model and associated concepts and terminology. Terms like tuple, attribute, domain and constraint were all clearly discussed. Relational Algebra joins and distributed database concepts were also explored well. While not the most popular question, it was generally answered very well.

**B5**

## GENERAL COMMENTS

This was a popular question, attempted by the vast majority of candidates. Nearly all of the attempts achieved pass level marks.

You are a **data architect** bidding for a new contract with a prestigious blue-chip organization. Part of the selection process is a technical interview. Answer the following questions from the interview panel.

a) Describe the defining characteristics of a *data warehouse* and how it differs in content and purpose from an 'OLTP' database system. You should use your own suitable examples and/or diagrams as needed.

**(10 Marks)**

b) Explain, the term 'ETL' with respect to data warehouses, taking care to highlight common problems or issues in each stage. You should use your own suitable examples and/or diagrams as needed.

**(10 Marks)**

c) Consider the phrase 'OLAP'. Explain what the term means, the underlying concepts involved, any associated benefits or limitations, typical applications and features along with any additional technical or implementation points you think appropriate to mention. You should support your discussion with suitable diagrams and/or examples.

**(5 Marks)**

## ANSWER POINTERS

A strong response will discuss and provide examples for:

- Cartesian Product
- Recursive Joins
- Natural Joins
- Inner Joins
- Left Outer Joins
- Right Outer Joins
- Full Outer Joins

## EXAMINERS' COMMENTS

The topic of data warehousing and its contrast with OLTP was generally covered very well. Likewise, the whole ETL process was very well described, with many candidates supplying well-annotated diagrams. OLAP was clearly a concept that most candidates were familiar with. Related issues such as data cubes, pivoting, dimensions, drilling down and rolling up were also strongly discussed. Overall, a very strong question for those who attempted it, with some obtaining maximum marks.

**END OF EXAM**