THE BCS PROFESSIONAL EXAMINATION
Professional Graduate Diploma

April 2004

EXAMINERS' REPORT

Advanced Database Management Systems

Question 1

1. Extensible Markup Language (XML) is becoming the most commonly used data format for storing semi-structured data. Describe a scheme for storing XML data in a relational database. Discuss the performance implications of your answer for an application that retrieves a given XML element (e.g. given an XML file which contains product details, retrieve information about a given product). **(25 marks)**

Answer Pointers

Most major commercial databases e.g. Oracle, DB2, QL Server and Sybase have schemes for storing XML in the database. Candidates may choose to describe one of these or devise their own method. For example, Oracle 9i uses a column type XMLType to store XNL as CLOBs. The advantage XMLType over a standard CLOB is that XMLType has a number of XML related functions (e.g.an implementation of XPath) associated with it. But this is not quite the same as reflecting an XML file as a set of tables since searching with SQL must be augmented with XPath expressions. Rather than disaggregating the data in to entities and mapping them to separate tables this mechanism stores XML files in their entirety. Query processing performance will depend on the mechanism by which the XML is placed in the relational database but in general performance will suffer. XML is inherently tree structured and relational databases are not generally suitable for tree traversal queries.

Examiner's Comments

Very few candidates chose to answer this question. The few answers that were submitted fell into one of two categories. Some candidates had no idea how to answer this question (no real grasp of XML) and were able to attract very few marks. The second category contained candidates who clearly did understand the issues related to the storage of XML. The average mark of this second category on this question was quite high.

Question 2

2. "We do not believe that the database language SQL is capable of providing a firm foundation for the future." (Hugh Darwen and Chris Date).

Critically evaluate this statement with reference to SQL's support for the relational model of data. **(25 marks)**

Answer Pointers

In this question, candidates are required to make the distinction between the standard implementation of the relational model via SQL and the relational model itself. The statement is contentious but made by significant figures in the database community and widely publicised. Candidates should understand that SQL breaks some basic relational model rules such as allowing duplicate rows. It also implements some aspects of the relational model in a clumsy way e.g. the use of null values. These mistakes appeared in very early versions of SQL. For compatibility reasons, accepted flaws in SQL have been carried through to later version of the standard. These then can make if difficult to add new features such as those envisaged for

object-relational databases. In spite of this, candidates should show an awareness of the universality of SQL in the relational world and the expectation that it will predominate in the foreseeable future.

Examiner's Comments

A very popular question with a number of good answers submitted. Variations of this question have appeared in the exam before and so it is likely that candidates were prepared for it. Candidates were often able to list the weaknesses of SQL without an apparent awareness of how they came about. As a consequence they often assumed that such weaknesses could easily be removed from the language whereas, in fact, for backward compatibility reason they are likely to be preserved in future versions of SQL.

Question 3

3. *a)* Explain what is meant by the terms *granule*, *schedule*, *permutable actions* and *serialisability*. Show how these concepts can be used to determine if two database transactions may execute concurrently. **(10 marks)**

b) Demonstrate how the principles you have described in your answer to part *a)* have been embodied in the following three concurrency techniques:

i) Timestamp algorithms
ii) Optimistic concurrency control
iii) Locking **(15 marks)**

Answer Pointers

a) A transaction is a series of actions on granules. A granule is the smallest part of the database on which the DBMS can operate. This might be a row, a table, some intermediate structure (page) or the whole database. Transactions carry out actions against these granules. Actions on granules are atomic. In some cases the order of these actions will not matter. For example, if two transactions only read data then the order those reads take place does not matter and the read action is said to be permutable. On the other hand if one transaction writes data another transaction reads then the point at which this takes place affects the outcome of the second transaction. Pairs of actions such as read, write; write, read and write, write are not permutable. Given two or more transactions running concurrently the operation on the database will consist of an interleaving of the ordered actions of each of the transactions. This is called a schedule. If there are two transactions T1 and T2 then those schedules which are equivalent to T1 followed by T2 or T2 followed by T21 will yield correct results. Such schedule are said to be serialisable. Serialisable schedules can be identified by constructing a precedence graph (based on the actions which are non-permutable and which therefore require to execute in a certain order). If there are cycles in the precedence graph then the schedule is not serialisable.

[10 marks]

b) Timestamp algorithms directly implement the concepts of permutable actions. Granules are stamped with a transaction identified that indicates the time at which that transaction started. If an earlier transaction subsequently tries to access that granule it is restarted. An improved timestamp algorithm discriminates between the different action read and write. [5 marks]

In optimistic concurrency control no action is taken until the transaction tries to commit. At the time commit takes place the actions of the transaction are examined with respect to all those transaction which have committed successfully since it began. If it is found to conflict with any of these transactions it is rolled back and restarted. [5 marks]

In locking transactions must acquire locks on granules they require prior to execution. Locks will indicate the type of access required (read or write). Consequently any transactions which have a

chance of carrying out non-permutable operations on the same granule are not permitted to proceed until such an occurrence is no longer possible. [5 marks]

Examiner's Comments
A popular question which attracted reasonable answers. Candidates, in general were able to answer part a) satisfactorily. As this was largely book work, this is not surprising at advanced diploma level. In part b), candidates generally offered answers which simply described the listed techniques but did not attempt to explain them. For full marks this was not sufficient. Full marks were only awarded to candidates who explained how the techniques ensured serialisibilty using the terms found in part a) of the answer.

Question 4

4. "An objective of a distributed database system is that to a user it should behave in exactly the same way as a non-distributed database system".

a) With the aid of examples explain why it is difficult to achieve the above objective in practice. **(10 marks)**

b) With the aid of diagrams describe the technology needed to co-ordinate and manage the physical distribution of data using 'replication'. Discuss the trade-offs that are needed to configure a distributed database application that supports replication. Apply your answer to a distributed database application with which you are familiar. **(15 marks)**

Answer Pointers
Database products such as Oracle, DB2, SQL Server support the physical distribution of data (at remote locations) without reliance of a centralised database. The measure of how far this distribution goes (ie does it behave and function exactly like a non-distributed system) is referenced by CJ Date who states 12 rules of compliance. These rules form the backbone to the answer and an understanding of the consequences of these rules should make candidates realise why truly distributed databases are hard to achieve in practice. The main points are trade-offs which compromise the notion that a distributed database should behave the same as a centralised database, namely:

- transactional integrity versus performance
- problems in maintaining a connected schema (a distributed data dictionary vs central one)
- global vs local query opimisation
- master – slave vs pure autonomy of replicated data (10 marks)

Part (b)
This part should be answered by looking at technologies offered by a particular database vendor that candidates at this level should be familiar with.
For example SQLServer replication offers a range of replication technology to support a range of options ie

- high transaction integrity traded off for less autonomy
- highly uncoupled databases with high latency and low transaction integrity.

The technology trade-offs also include the role of 'middleware' and the role of middleware imposes on transactional integrity. Replication technology provides middleware that does one job managing the transfer and synchronising shared data physically distributed across multiple servers. The 'middleware' also contains the replication agent software and distributed components. For example SQL Server supports the two ways of distributing a database through federation, which distributes the processing load of the database server ('scaling out') and replication which distributes the data across different physical locations ('scaling up') . (15 marks)

Examiner's Comments

A popular question but generally not very well answered. A crucial aspect of this question was to associate Dates rules (which appear in all of editions of his textbook) to the problem of physically distributing data. Few candidates made or even noticed this association. Instead answers seem to fall fowl of regurgitating notes on distributed database architectures. As for as part b), many candidates seem to show a lack of system-wide skills in how replication is configured, perhaps this is a weakness of textbooks and candidates may find it useful to read literature from database vendors on how distributed databases are configured in real world applications, particularly where the physical distribution of data is a necessity. There were a group of candidates from one centre who gave some very good answers and it is worth noting that these candidates seemed to have experimented with SQLServer replication on a local servers.

Question 5

5. *a)* Explain the differences between Object Oriented and Relational data models in the way that they model the following:

i) Object/Tuple identity
ii) Object/Tuple relationship **(8 marks)**

b) Many database applications have constraints called 'business rules', which are programmed to prevent a database update taking place if a rule is violated.

Assume a RDBMS is deployed on the server side of an n-tier client-server platform and the following business rules are required.

Business Rule#1*: From a library database:*
Borrowers with status = 'student' cannot have any further loans if they have currently borrowed 5 items already because 5 is the loan limit for this type of borrower.

Business Rule#2*: From an exams database:.*
Students are assessed on a course by taking 4 exams. Exam marks for a particular student are entered one after the other, if the mark entered is less than 30% then an overall grade of FAIL is returned and no further marks can be entered for this student. When all 4 marks are entered then the average mark is calculated and a grade of PASS or FAIL returned. For a PASS the average mark must NOT be less than 40%. Otherwise the grade given is FAIL.

i) Explain how each business rule could be programmed to check for compliance following user interaction on a client interface. Illustrate your answer with sample code or pseudocode.

ii) Discuss the factors that influence the decision as to whether business rules are implemented in middleware or on the database server in an n–tier client server platform. **(17 marks)**

Answer Pointers

Object identity is intended to maintain persistence of object identity that will outlive changes in technology and infrastructure. Object identity can be time-stamped thus allowing copies of objects to exist but still retain uniqueness internally at the system level.

In tuple identity the focus is on primary keys (or entity integrity) taking on the role of uniqueness. Clearly identity is systematically linked to the DBMS and thus persistence exists within the lifetime of the application or operation of the DBMS. Problems occur when tuples are duplicated for example during replication. An identity operator is used to create a globally unique identifier but this implementation is still DBMS specific unlike Object identity.

Object/Tuple relationships are handled differently in OO and Relational models. Referential integrity is not supported implicitly in the OO model but it is in the Relational model through Primary Key-Foreign Key constraints. Relationships in the OO model can be modelled on

inheritance, aggregation and simple instances connected via a mechanism similar to linked lists. The Relational model is much simpler and uses the basic unit Primary Key-Foreign Key principles to represent the more complex structures of inheritance and aggregation.

Part b)
SQL provides a range of techniques to ensure application integrity is maintained through compliance of business rules these include:

- triggers
- user defined functions
- stored procedures

A trigger should be written for example to check the INSERT operation on a hypothetical table (Loans when a member request a loans that will exceed their loan limit. A trigger will 'fire' when the INSERT is detected and will run some code that collects the state of loans for this borrower and performs some tests. If these tests are negative a business rule is violated and the recovery is made (usually a rollback for an After trigger). There are stored procedures that may run on a routine basis to check that the member does not have any outstanding fines or the book is not reserved by someone else.

Examiner's Comments
A very popular question with a number of good answers submitted for part a). The main problems surfaced in part b) where candidates still think of relational constraints as the only means by which application integrity is maintained.

The primary and foreign key constraints; check constraints are NOT sufficient to handle the constraint checking imposed by business rules. For example loan limit = 5 cannot be implemented as a check constraint which some candidates seemed to believe.

Very few candidates could even appreciate the dilemma of how business rules are implemented and therefore could not follow-on and provide any sensible recommendations or coding examples. Those candidates who submitted code had misread the question and hence wasted a lot of time reproducing large amounts of irrelevant code with little or no functionality (ie interspersed with SQL). In both these business rules the examiner expected pseudo-code with the most important content being a liberal interspersing of SQL code and not pages and pages of irrelevant code.