BCS THE CHARTERED INSTITUTE FOR IT

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

ADVANCED DATABASE MANAGEMENT SYSTEMS

Friday 27th March 2015 - 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

EXAMINER'S GENERAL COMMENTS

Just over a third of candidates attempted this question. The average mark was slightly below expected as the subject matter is frequently covered in this exam. There was a good range of marks and evidence of increasing knowledge of this emerging application of database technology to the WWW.

In recent years there has been a trend to represent traditional relational data in a semi-structured data format such as XML.

a) List and explain three of the main motivations for this trend over recent years. **(6 marks)**

ANSWER POINTER
Consistent, portable standardised transport mechanism over WWW
Richer non-relational data formats such as GIS
Volume of data in a simple format i.e. effectively text files
Extensive support derived on WWW from w3 standardisation

EXAMINER'S COMMENTS
Candidates displayed fairly good knowledge overall in this part with some good insight and working knowledge of XML based databases.

b) State an application and explain why the stated application warrants the need to store and process semi-structured data. **(6 marks)**

ANSWER POINTER

One example from usual applications eg Document centric databases to GIS (geographic information systems) where it is more natural to express hierarchical data or text formatting in documents. Such applications consume vast amounts of semi-structured data that are more query intensive rather than transaction (ie many updates) intensive. These applications usually represent presentation logic close to actual data format.

EXAMINER'S COMMENTS

The term 'application' was misinterpreted by some candidates as technology such as Xpath and XSLT.

c) Explain the support required for storing semi-structured data in a relational database. Hint: use a simple example such as a file that contains product details. **(7 marks)**

ANSWER POINTER

First of all a schema is required. The syntax of the schema is not important rather an appreciation of the active xsd elements in the schema

```
<xs:element name="product">
  <xs:complexType>
    <xs:attribute name="prodid" type="xs:positiveInteger"/>
  </xs:complexType>
</xs:element>
```

The next task is to use a translator such as XSLT that transforms an XML file into tables. Alternatively source XML can be stored natively in a column of the database,

EXAMINER'S COMMENTS

The 'mechanics' of translation and manipulating are very much the same, implemented by many technologies. The best answers were from candidates who could produce a step by step or walk through of the translation process applied specifically to an example such as a product file.

d) Discuss the performance implications for retrieving a particular XML element from a XML file. **(6 marks)**

ANSWER POINTER

This depends on the method used to store XML data. Most RDBMS have schemes for storing XML either as a column type (XML) or text. The advantage over standard text (CLOBs) is that a number of high performance implementations of Xpath can be associated with it. As XML is tree structured and RDBMS are not, performance is better for 'tree-traversal like' operations but not for sequential scans or qualified /aggregate queries.

EXAMINER'S COMMENTS

Again the best answers were those that applied examples showing how high performance is achieved using say Xquery or Xpath. Again only a small number of candidates had sufficient working knowledge of these languages and hence marks were generally low.

A2

EXAMINER'S GENERAL COMMENTS

Just over a quarter of candidates attempted this question. The average mark was lower than expected. Candidates should expect at PgD level not to simply recall and regurgitate facts when answering a question. Candidates therefore seem reluctant to attempt questions where reflection and judgement is required in formulating an answer to a problem domain (in this case a simple scenario and business rule). Both the scenario and business rule were intended to be clear and concise.

Because of the increasing storage capacity of memory, a typical database application can now cache most of the application's data requirements in internal memory.

a) Explain the concept of data persistence and explain the impact on data persistence given the above statement. **(8 marks)**

ANSWER POINTER

Data persistence is a term to describe how data is maintained in a state that can be recovered and data integrity maintained over a long period of time. All DBMS's support persistence by their very nature but they are optimised to reduce disk I/O by holding data in memory particularly for query intensive/long duration transactions. The impact on data persistence depends on the type of application whether OLTP (on line transaction processing) requiring frequent changes to database state to be committed and checkpointed (to ensure data integrity) or whether the type of application is OLAP based, with large amounts of data cached whilst the user works on it and saving the state is infrequent.

The duration of transactions between commits also has to be considered, whether for instance an OLTP application operates in real time and user controls the saving and

retrieval of database states (such as CAD design); or whether there is a high level of concurrency such as on-line shopping whereby traditional methods of disk intensive commit based protocols apply.

EXAMINER'S COMMENTS

Although most candidates could explain data persistence, many answers were too shallow for this level. It was disappointing that many candidates could not apply knowledge of many well established development techniques that utilise memory based databases in web applications, such as in ASP.NET where objects are directly mapped or transformed from tables. These tables are created acting in a non-persistent way between database connect/disconnect statements.
Also it was expected that candidates at this level should read about future database technologies that support no-disk I/O such as 'TimesTen'. These database products are unlike traditional disk-optimised relational databases such as the Oracle Database, DB2, Informix or SQL Server, whose designs must contain algorithms that attempt to minimise disk accesses.

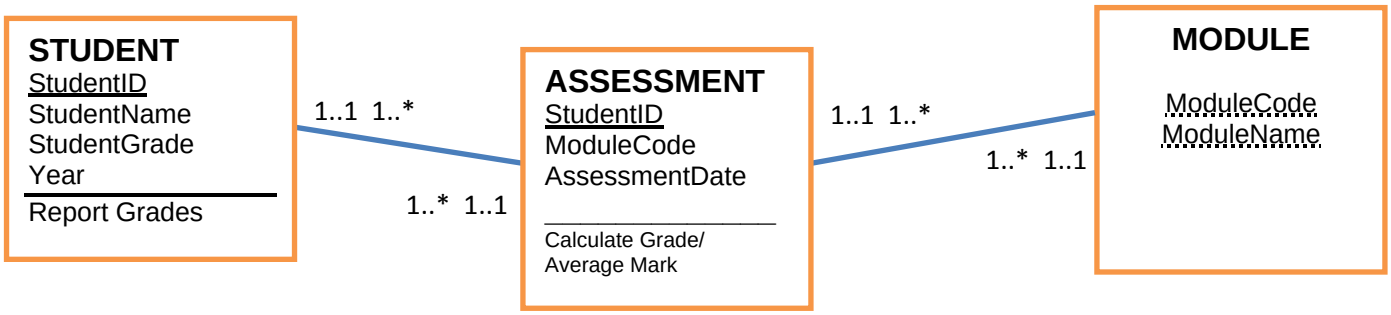
b) Consider the following scenario that describes the processing of examination results at a college on a database that holds information on student assessment.

Students are assessed on a particular course by taking 4 exams. Each exam is the only assessment for a module on which they have enrolled. The students from different courses share the same module. Exam marks for a particular student are entered in sequence. A report is generated showing the end-of year assessment results with the following column headers :-
Student_ID, Student_name, Date_Assessed, Average_mark, Grade

i) Using this information derive a simple CLASS (Object Oriented) model using a defined notation (for example UML) **(5 marks)**

ANSWER POINTER

Using UML Class diagram notation



EXAMINER'S COMMENTS

This presentation of a Class model is required to identify the underlying tables and which methods apply - in other words the data requirements needed to support the business rule that follows. Again it was a bit disappointing that many candidates struggled with applying a familiar and simple scenario to an OO model. Care must be taken in defining a suitable notation.

ii) A database trigger could be used to implement the following business rule.

Business Rule:-
If the mark entered is less than 30% then an overall grade of FAIL is recorded. When all 4 marks are entered then the average mark is calculated and a grade of PASS or FAIL recorded. For a PASS the average mark must be 40% or more with no single mark less than 30% otherwise a FAIL is recorded.

Explain with the aid of sample data and pseudo-code how this could be achieved and discuss the advantages and disadvantages of using triggers in this way.

(12 marks)

ANSWER POINTER

Candidates are expected to show an understanding of the program logic via pseudocode allowing for a range of alternative solutions. Implementations of rules or conditional constraints normally require procedural code implemented in stored procedures and triggers. Answers related to application/logic or middleware such as Java PHP, ADO.NET are invalid as they do not run on the database server. Therefore answers cover the use of trigger and stored procedures with the emphasis on database state before and after an update (affecting the business rule) takes place. A trigger should be written for example to check the INSERT operation in a hypothetical table Assessment derived from the Class model. A trigger will then fire when the INSERT operation is detected and will run some code that collects the state of the assessment either before or after (before or AFTER TRIGGER). Conditions are tested to see if the mark is less than 30 and if so the transaction that did the insert is rolled back (case for an after trigger) Trigger code should be implementation neutral so pseudo-code asked in the main body but the header should describe the type and role of the trigger. A similar trigger could be used to compute the results or a stored proc could be used.
A specific implementation is not required due to the various ways that recovery and rollback occurs if the business rule is violated.

Pros and Cons of triggers

Pros: Application logic is close to user interaction. Captures and checks intentions much easier eg by validation rather than overload the database I/O with possible unnecessary activity. Program logic close to the data it affects. Easier to reduce I/O and impose transaction integrity. Performance can be optimised more readily using statistics and query optimisation. Basic logic built on SQL so extends SQL with richer facilities and thus a smaller range of expertise required.

Cons: Trigger code is event driven and very hard to test and handle exceptions. Cascade of multiple triggers can cause deadlock for example. Triggers have limited program functionality therefore suitable for simple decision logic but falls down when excessive say use of triggers are active. The real time situation of triggers firing and interleaving can be difficult to test and maintain.

Therefore an alternatives are sought such as middleware supporting OO features with much richer and disciplined implementation frameworks that are easier to test and reuse and largely independent of database server and platform.

Changes to the database for example move to a new server / data structure can impact on existing trigger code.

EXAMINER'S COMMENTS

Most candidates had little difficulty in defining an algorithm and pseudocode. But in most cases the code samples lacked an explanation of the activation mechanisms of triggers. Triggers are specific database procedures that occur following a database event (such as INSERT) on a particular database table. There was a better appreciation of pros and cons of triggers but generally answers were quite shallow and superficial.

A3

EXAMINER'S GENERAL COMMENT:

48% of candidates attempted this question which had a high pass rate of 79%

(a) Consider the following three linked tables that contain information about employees and the projects they work on:

```
employees (empID, name, salary)
project (projNbr, title, budget)
```

```
workload (empID*, projNbr*, duration)
```

Consider the following query:

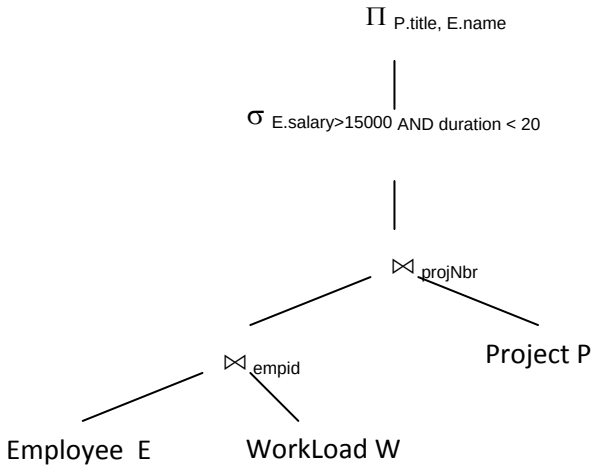
```
SELECT P.title, E.name
FROM employees E, project P, workload W
WHERE E.empID = W.empID
AND P.projNbr = W.projNbr
AND E.salary > 15000
AND W.duration < 20;
```

(i) Draw an initial relational algebra tree for the above query. **(4 marks)**

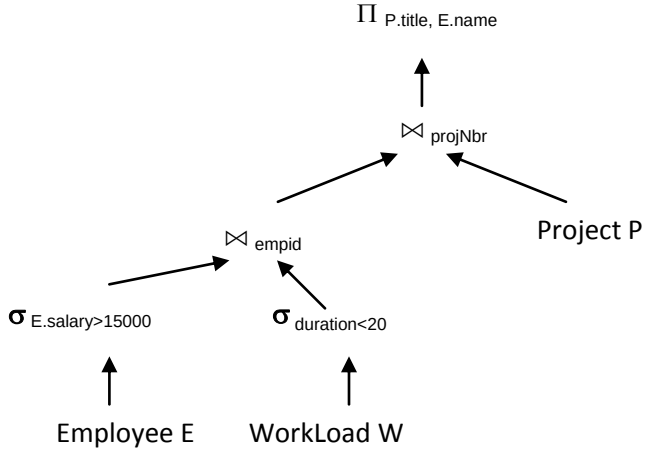
(ii) Apply a series of transformations to the tree obtained in part (i) to make the query more efficient. Discuss each step and state the heuristic used. **(12 marks)**

ANSWER POINTER

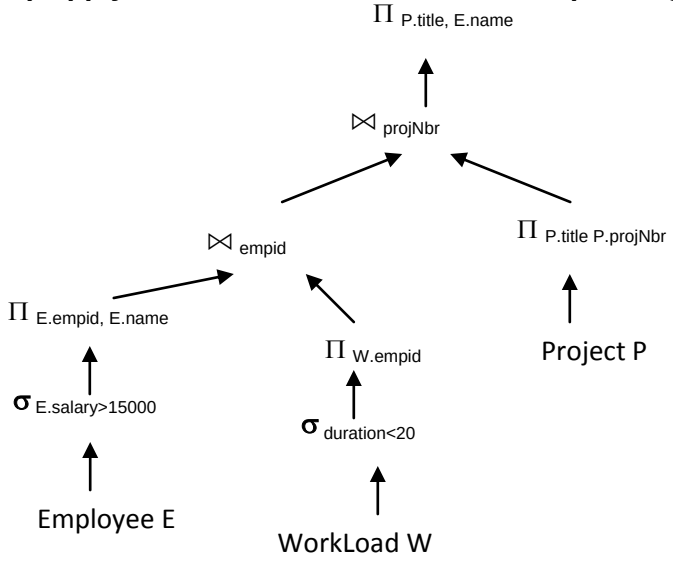
(i) The initial relational algebra tree is as follows: (4 marks)



(ii) Step 1: (6 marks) Apply the heuristic that recommends pushing the selections down the tree:



Step 2: (6 marks) Apply the heuristic that recommends pushing the projections down the tree:



EXAMINER'S COMMENT

Most students managed to correctly optimise the query tree. A few confused the symbols used for projection, selection and join.

(b) Suppose you have the following table:

Employees (empID, lastName, firstName, salary)

Suppose that the most frequently executed queries on this table are as follows (the "?" indicates any given value):

- `SELECT * FROM Employees`
  `WHERE firstName = ? AND lastName = ?`
- `SELECT * FROM Employees WHERE lastName = ?`
- `SELECT empID FROM Employees ORDER BY salary`

Suppose there are four indexes as follows:

- Index1 (`empID`)
- Index2 (`firstName`)
- Index3 (`lastName`)
- Index4 (`salary`)

(i) There is a need to improve the performance of all three of the above queries. Select **two** indexes from the above list that if built and available will result in an improved performance being obtained for all the above queries. Explain your

answer. **(6 marks)**

(ii) Describe three disadvantages of using indexes. **(3 marks)**

ANSWER POINTER

(i) Index3 and Index4 should be selected.
Index3 can be used to enhance the performance of query1 and query2: both have lastname in the WHERE clause. (3 marks)
Index4 can be used with query3 given that an ordering on the attribute salary is going to be achieved as a by-product of indexing. This will avoid running a separate expensive sorting algorithm (3 marks)

(ii) Disadvantages are related to the overhead of using indexes and can include things like:

- **Whenever an insert/update/delete operation occurs on a table record, a corresponding operation would need to be performed in the index**
- **Increase in disk space to store indexes**
- **Possible performance degradation if query optimiser considers all available indexes before selecting an optimal strategy.**

EXAMINER'S COMMENT

Answers suggest that some candidates are not clear about when an index can be useful for a query. However, most are able to recognise the overhead introduced by an index as a disadvantage.

Section B

Answer Section B questions in Answer Book B

B4

EXAMINERS OVERALL COMMENTS

An extremely popular question attempted by almost all candidates (98%), the vast majority achieving pass level marks (90%).

(a) *Using your own suitable examples and diagrams*, explain in your own words what the following database concepts mean:

(i) Database Transaction **(3 Marks)**
(ii) ACID Properties **(3 Marks)**
(iii) Isolation Level **(3 Marks)**

ANSWER POINTER

A transaction is a logical unit of work (comprising one or more operations) that transforms a database from one consistent state to another consistent state (or very similar). Students should make it clear that a transaction is a concept commencing with a DML operation and concluding with either a rollback or commit and is not the same as an individual SQL statement or query. Students should then move onto ACID as specifying atomicity, consistency, isolation and durability - each aspect must be named and explained in detail. Bonus marks for a clear worked example or well annotated diagram.

EXAMINER'S COMMENTS

Just about every candidate gave a textbook discussion of what a database transaction is and what the ACID properties were all about – so heavy marks were acquired by most. Regarding isolation levels, this was less well addressed and indeed, several candidates skipped it completely. But overall a good question for most. Some candidates gave good-quality annotated diagrams but many did not.

(b) *Using your own examples and suitable diagrams*, discuss the following transaction-processing concepts:

(i) COMMIT **(3 Marks)**
(ii) TWO-PHASE COMMIT **(3 Marks)**
(iii) ROLLBACK **(3 Marks)**
(iv) SAVEPOINT **(3 Marks)**

ANSWER POINTER

COMMIT as a saving operation (of the complete transaction), ROLLBACK to undo the whole transaction, SAVEPOINTS as transaction partitioning concepts whereby a large transaction can be sub-divided down into smaller units using embedded labels and how the DBMS can rollback to a named savepoint rather than undoing the whole transaction. Candidates should then go on to describe the need for two-phase commit in a distributed database environment and how it constitutes the 'voting' phase (the 'are you ready to commit' message) and the 'decision' phase (the 'commit now' message). Candidates should also cover the concepts of *global transactions*, *localized sub-transactions*, *transaction coordinator site* (the transaction initiator site), *participant sites*, the need to pass *messages* between sites, the use of *timeouts* to avoid unnecessary blocks or delays, the possibility of a participant site issuing a *local abort* – thus forcing a *global abort* to be issued and the need for unanimous *local commits* before a *global commit* is circulated – all ensuring that *data integrity* and *ACID rules* are satisfied.

EXAMINER'S COMMENTS

Again, generally well answered – certainly regarding COMMIT, ROLLBACK and SAVEPOINT. The comments on TWO-PHASE COMMIT were less successful, with often vague or wrong answers. This was one situation were a well-annotated diagram would have been useful to clarify the distributed nature of the essential problem and the two distinct stages involved – but sadly very few students supplied such a diagram. Those that answered it well, even those without a diagram, did make it clear how the 'voting' and 'committing' stages worked.

(c) For each of the following transaction control terms, write a *single sentence* (no need for extended responses, examples or diagrams) explaining the key concept.

(i) Cascaded rollback **(1 Mark)**
(ii) Optimistic locking **(1 Mark)**
(iii) Pessimistic locking **(1 Mark)**
(iv) Checkpoint **(1 Mark)**

ANSWER POINTER

Cascaded rollback - when transaction T1 fails and induces a rollback which in turn causes other transactions - which were dependent on the success of T1 – to likewise fail and be rolled back. Optimistic locking – based on the assumption that inter-transaction conflict is rare so individual transactions are allowed to proceed unsynchronized and are only checked for conflicts at the end – just before commit. Useful in low data contention environments because it avoids the overhead of locking/waiting for unlocking but inefficient if data conflicts are common as transactions will have to be repeatedly restarted. Pessimistic locking

– assumes a high degree of inter-transaction conflict and locks up all data resources ahead of access immediately – even if other transactions never try and access them. Saves re-running transactions in highly contentious environments but this 'belt and braces' approach can induce overhead in locking/releasing data resources that were never in conflict in the first place. Checkpoint – A point of synchronization between the database and the transaction log file (journal) that allows easy identification of which transactions need to be re-processed (redo/undo) in the event of database recovery being needed.

EXAMINER'S COMMENTS

There was often a disappointing end to this question, with many vague or superficial responses – particularly for the two locking terms. There was some confusion between checkpoints and savepoints and a significant number of candidates missed the essential nature of cascaded rollbacks – thinking it was the same rolling back process (for the same transaction) but executed in a staged manner – rather than it relating to the rewinding of several distinct transactions.

B5

EXAMINERS OVERALL COMMENTS

A very popular question attempted by almost all candidates (89%), the vast majority achieving pass level marks (91%).

(a) Describe the key characteristics of a *data warehouse* and how it differs in content, structure and function from an *on-line transaction processing (OLTP)* database. You should support your discussion with suitable diagrams and examples **(10 Marks)**

ANSWER POINTER

Key points should include…

An OLTP database is designed to support low-level, day-to-day transactional data with a view to satisfying the core business activities by junior to mid-level staff. By contrast, a data warehouse is all about long-term strategic decision making by senior management. The data warehouse is commonly defined by four attributes: *subject-oriented* (customer, product etc.), *integrated* (uniform data drawn from multiple sources), *time-variant* (historical data means a series of snapshots of data over a time period) and *non-volatile* (data is always added, often aggregated and usually not deleted or modified). An OLTP database will, for efficiency reasons, not hold historical data and is application/business-oriented (customer invoicing etc.) It is the difference between a low-level routine transaction-focus and a high-level ad-hoc, analysis focus. Quality diagrams and clear examples gain bonus marks.

EXAMINER'S COMMENTS

Generally well done with the vast bulk of students clearly (and at some length) describing all the points raised in the marking scheme – hence the generally high scores. The stronger students usually supplemented their narrative with quality annotated diagrams. A good question for nearly all who attempted it.

(b) For each of the following items, 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.

(i) OLAP *(hint: think different implementations of OLAP, SQL and OLAP, aggregation etc.)* **(5 Marks)**

(ii) Multi-Dimensional Data *(hint: think what each dimension could represent, roll-up, pivoting)* **(5 Marks)**

(iii) Data Mining *(hint: patterns and prediction, techniques to identify these, data preparation, tools)* **(5 Marks)**

ANSWER POINTER

OLAP (On-Line Analytical Processing) is the dynamic synthesis, analysis and consolidation of large volumes of aggregated multi-dimensional data (or similar). Key features include aggregation, time-series analysis, drill-down queries, highly complex 'what if' and 'why' type queries (as well as the more traditional OLTP-type 'what', 'when' and 'who' queries) across multiple views (dimensions) of the data over extended time periods. Applications could include budgeting and financial modelling, sales analysis & forecasting, market research etc. The better students may go onto discuss Codd's twelve rules for OLAP tools, the different categories of OLAP – MOLAP, ROLAP and HOLAP (multi-dimensional, relational and hybrid respectively) including a brief discussion of the key features of each. Finally, a few comments on OLAP extensions to SQL could be included such as the ROLLUP and CUBE functions (within the GROUP BY clause) and RANK, DENSE_RANK functions and 'windowing' calculations on aggregates.

Multi-Dimensional Data begins with the traditional two-dimensional relational table before moving onto three-dimensional data cubes and, logically, four dimensions and more (for example dimensions could be time, location, product, staff) with each cell in the 'cube' representing sales figures. To maintain acceptable query performance as the number of dimensions rises, 'rolled-up' or pre-computed (consolidated or aggregated) data may be stored that in turn may be drilled down for finer detail as needed (for example, time could be aggregated to years but drilled

down to quarters, months, weeks and even days worth of sales). Data may also be 'sliced & diced' (pivoting). The role of dense/sparse data in different cells and the role of data compression could be covered.
Data Mining – The process of identifying patterns & trends and thus extracting previously unknown, actionable, valid and valuable information from large data sets for commercial or other advantage (or similar). Typical applications could be identifying customer buying patterns, detecting patterns of fraudulent credit card use, analyzing insurance claims/frauds etc. Techniques include *predictive modelling* (using supervised learning with training/testing stages), *link analysis* (based on establishing associations or affinities between different data items) and *deviation detection* (using statistical or data visualization techniques to identify 'outliers' in the data that defy 'normal' or expected behaviour – thus leading to new, unexpected knowledge). The better students could also cover the key issues around DM tools such as data preparation (data cleaning, handling missing data etc.), selection of data mining algorithms (how it relates to the training/testing stages and how it handles 'noise' in the data), and scalability & performance (use of parallel processing and other optimization issues).

EXAMINER'S COMMENTS

Again, a very good performance by the vast majority of candidates. Answers were detailed, extensive and normally accompanied by many well annotated diagrams and case studies. Just about all students scored heavily here. Unlike the previous database transaction question, it was perfectly clear that when it comes to data warehousing and associated techniques like OLAP and data mining, there was a good understanding of core concepts. A strong question all round.