BCS THE CHARTERED INSTITUTE FOR IT
BCS HIGHER EDUCATION QUALIFICATIONS
BCS Level 5 Diploma in IT

September 2013

EXAMINERS REPORT

Database Systems

SECTION A

Question 1

General Comments (Q1)

This question was the most popular, having been attempted by over 80% of the candidates. Around 70% achieved pass level marks.

Answer pointers part a:

(i)
(QuestionNo, QuestionText, CorrectAnswer, CandidateAnswer) is the repeating group.

Put the un-normalised table into 1NF as follows:

Candidate(CandidateID, Name, Address)

Result(CandidateID*, QuestionNo, QuestionText, CorrectAnswer, CandidateAnswer)

(ii) Partial Dependencies:
QuestionNo → QuestionText
QuestionNo → CorrectAnswer

Candidate(CandidateID, Name, Address)

Question(QuestionNo, QuestionText, CorrectAnswer)

Result(CandidateID*, QuestionNo*, CandidateAnswer)

(iii) There are no transitive dependencies, so the tables are in 3NF

Examiner's Comments: Although most candidates understand how to normalise tables, there is lack of understanding of the definition of partial and transitive dependencies. This was evidenced by the fact that many candidates can produce tables in 3NF directly from the un-normalised form, but find it difficult show the normalisation process from 1NF to 2NF to 3NF.

Answer pointers part b :

(i) The table violates 2nd Normal Form (1 mark) because there are two partial dependencies: StudentID → StudentName (1 mark) and ModuleID → ModuleName (1 mark)

(ii)

Student(StudentID, StudentName)

Module(ModuleID, ModuleName)

Results(StudentID*, ModuleID*, Grade)

Examiner's Comments: Most candidates managed to answer part (ii) correctly. However, a few correctly identified 2NF as the violated normal form. Candidates need to make sure they have a good understanding of the various normal forms.

Answer Pointers and Marking Scheme: 5 marks for at least 5 main characteristics and their description/contrast with File-based systems, such as:

Data independence, Control of data redundancy, Data consistency, Sharing of data, Improved data integrity, Improved security, Enforcement of standards, Economy of scale, Improved maintenance, Increased concurrency, Improved backing and recovery services.

Examiner's Comments: Most candidates have a good awareness of the main characteristics of database systems in contrast with file-based systems. However, candidates need to be careful not to refer to file-based systems as "paper-based" systems.

Question 2

General Comments (Q2)

This was a popular question with 76% of candidates attempting it and of those 50% achieved pass level marks. There was a range of attempts from very reasonable to very poor. But very few candidates read the question regarding the notation many simply ignored the instruction and produced a variety of notations but essentially this was disregarded as the question was intended to test an understanding and conceptualisation of a scenario requiring a database solution. It is felt that the style of this question works as it gives the candidates a chance to shine on a typical scenario without being too precise on notation.

Answer pointer part b :

There are a number of possible scenarios involving job, depot, transport unit, load and product entities

Examiner's Comments :

Very few candidates stated assumptions.

Answer pointer part b :

Eg A Depot holds 1 or many Transport Units but a Transport Unit belongs to 1 and only 1 depot

Examiner's Comments :

Generally good answers, though some candidates did not follow the example and some even echoed the example given for some reason.

Answer pointer part c :

Credit was given for correct columns in tables that related to the candidate's entities and also the identification of primary and foreign key columns.

Examiner's Comments :

Poor attempt many candidates misread the question and produced table definitions which did attract some marks. The tables should have been at least in 2NF but clearly many candidates could not and did not derive tables from the ER model they produced in a) despite this being stated in the question. This part was the most disappointing aspect of the candidates' performance.

SECTION B

Question 3

General Comments (Q3)

Popular question but poorly answered. It was worrying to see poor knowledge of standard JOIN operations in SQL. However if the code is unfamiliar, it is more difficult to derive the output from SQL code than formulating the code to generate a specific output. This does not fully explain the quality of the candidate answers.

Answer pointer part a :

Query 1

```
Mname     budget
--------- ------
Ashforce    NULL
Bass      675.00
Mission   250.00
Vallance  348.00

(4 row(s) affected)
```

Query 2

```
Totalbudget Mname
----------- ---------
       NULL Ashforce
    1350.00 Bass
     250.00 Mission
     348.00 Vallance
```

Query 3

```
EmployeeID EmployeeID membership
---------- ---------- ----------
E3         E5         Bass
```

Examiner's Comments :

Query 1

Poor overall knowledge of OUTER JOIN was in evidence. The t.* means include all teams (t is alias for teams). This, and the meaning of distinct, caused a problem.

Query 2

Many candidates failed to see the effect of the SUM aggregate operator in the case of Bass resulting in a total budget of 675.

Query 3

Very few candidates could work out this resultset. There were many non attempts and many candidates stating they thought it would not produce any output, i.e. non computable. Candidates did not recognise the nature of the self join.

Answer pointer part b :

Query1: OUTER JOIN (RHS) where all values in the RHS of the designated join columns are included whereas the LHS are eliminated if they do not match.
Query2: INNER JOIN – the usual type of JOIN
Query3: SELF JOIN a table that JOINS to itself to achieve the functionality of hierarchical data sets.

Examiner's Comments :

Very few candidates recognised the self join and instead stated that it was a full outer join or a natural join - which is true but of a more specific type. The other types of JOIN were generally recognised in and reasonably described.

Answer pointer part c :

```
QUERY 2 amended for part c) to get desired result

SELECT ISNULL(AVG(budget),0) as budgetperEmployee, budget , Mname
FROM employees as e
INNER JOIN teams as t
ON e.membership = t.mname GROUP BY budget,Mname

-- note that isnull returns 0 where null value occurs
-- when computing average

-- ALTERNATIVE

SELECT ISNULL(budget/Count(*),0) as budgetperEmployee, budget , Mname
FROM employees as e
INNER JOIN teams as t
ON e.membership = t.mname
GROUP BY budget,Mname
```

Examiner's Comments :

Poor attempts overall with less than 10% of candidates submitting correct answers. Some candidates recognised that the ISNULL operator was required, but many simply recast query2 with little or no amendment.

Answer pointer part d :

NULLs are often ignored by aggregate functions. Counting values in a column for instance will not include NULLs in the count

Examiner's Comments :

Reasonable understanding of NULL values and some candidates even mentioned the use of ISNULL and NULLIF operators to overcome the problem and pitfalls.

Question 4

General Comments (Q4)

This was not a popular question with 30% of candidates attempting it and of those, only 30% achieved pass level marks

Answer pointer part a :

This will be marked holistically with extra marks for good examples and clear diagrams but good candidaates should cover: SELECTION, PROJECTION, JOIN, DIVIDE, CARTESIAN PRODUCT (TIMES), UNION (and maybe UNION ALL), INTERSECTION AND MINUS (DIFFERENCE) operators. The first four were developed specifically for relational databases by Dr. Ted Codd while the rest originate from mathematics (set theory). For each, a Venn diagram and suitable RA example is expected.

Answer pointer part b :

Five marks apiece. Each marked holistically but good examples and diagrams gain extra marks. First point covers how the SQL CREATE TABLE statement supports RM concepts like tables (relations), columns (attributes), rows (tuples), data types & domains, primary keys (identifiers and unique identification of tuples), foreign keys, constraints (NOT NULL, UNIQUE, CHECK) etc. Second point is all about how SQL implements operations like SELECTION (row-filtering via WHERE clause), PROJECTION (column-filtering via SELECT clause), the various types of JOIN (including how RA & SQL handle the duplicated and common columns used to make the join), implementation of CARTESIAN PRODUCT (TIMES) as an 'unqualified join' with no WHERE clause, UNION, INTERSECT and MINUS (DIFFERENCE) and the role of intermediate queries (plus the role of UNION COMPATIBILITY) etc. The third point refers to non-relational constructs such as SQL support for object-oriented structures (objects, methods, constructors, nested tables, varrays etc.), SQL and XML, SQL and other novel data types and data structures.

Examiner's Comments :

It was surprising how few candidates actually attempted Q4 (84 answers) – considering the core nature of the knowledge being assessed. Of those who did, most made a reasonable attempt at part (a) with many giving good accounts of the various RA operators and several knowing the correct symbols for each operation. The use of Venn diagrams was always good. Sadly, too many candidates did not follow this good work with clear concrete examples based either on the supplied relation or one of their own. They clearly knew the concepts but did not apply them with

specific examples. Few bothered to differentiate between those RA operators coming from set theory and those developed specifically for the relational model. So overall, a good question if attempted.
As for part (b), this was almost universally ignored - even by those who did part (a) – or if attempted, it was done very poorly. Only a handful of candidates grasped the ideas mapped out in the marking scheme (and were well rewarded) with most simply describing SQL itself (as opposed to explaining how SQL implements and captures RA operations such as selection/projection/join etc). They obviously knew SQL statements but could not relate those statements to RA concepts (for example, how the SELECT keyword implements the projection operation or how the WHERE clause implements the selection operator). Most did very badly on part (b).
This question tended to be avoided by large numbers of candidates, while those who did attempt it did quite well on (a) but extremely badly on (b). A tiny number did very well on both parts.

Question 5

General Comments (Q5)

This was a popular question with 65% of candidates attempting it and of those, 44% achieved pass level marks. Candidates tended to regurgitate standardized database responses and diagrams with most not reading into the questions and thus moving on from that standardized response. There was also a strong tendency towards generalized (and often vague) comments.

Answer pointer part a :

This question is essentially about how a form relates to the external level of the ANSI-SPARC model as a means of providing one (of possibly many different) user views onto the underlying conceptual level and the database it logically describes. A good diagram is essential. The better candidates will then go on to discuss SQL views as a programmatic equivalent of a form in the sense that the defining logic of the view provides the same data to the end user – albeit not in a graphical way.

Examiner's comments :

Part (a) was well answered on the ANSI-SPARC 3-level architecture but most candidates stuck to the textbook descriptions (and diagrams) and never really ventured into discussing the place and role of forms and other non-GUI interfaces played within that architecture. Most candidates got about half marks or just slightly better.

Answer pointer part b :

At the form level (front-end) the data can be validated using on-form logic and by avoiding invalid data in the first place by using drop-down lists to ensure only pre-validated entries can be chosen, radio buttons to ensure only a single (valid) option is selected, double-entry of key fields like passwords to rule out mistyping, automatic totaling of numerical data, on-form calendars where users can click on a given date, labels at side of each field with an example, on-form help

button, highlighting which fields are mandatory via an asterisk etc. - thus avoiding sending erroneous data to the back-end database and wasting a return journey over the network. So form level validation is simple but efficient.
At the database level (back-end), much more complex data validation – that requires access to the database tables or the data dictionary perhaps - can be employed. This necessitates a round trip to the server but it does mean that complex business logic (written as table constraints or SQL triggers) can be enforced. So database level validation is vastly more powerful.

Examiner's comments :

Part (b) addressed data validation and while most candidates raised valid *general* points about the role and importance of data validation, many missed the core issues laid out in the marking scheme – namely the different on-form techniques and how trapping errors upfront saved traffic and so aided efficiency. Some candidates did catch on and gave detailed explanations of radio buttons, drop-down lists, calendars etc and their role in combating erroneous data – but they were in the minority. Regarding database (back-end) validation, very few mentioned triggers, stored procedures or functions – although many did elaborate well on domain/entity/referential integrity issues.

Answer pointer part c :

The front-end web form, running inside a web browser (client), is responsible for data presentation and some simple data validation, the back-end DBMS is responsible for all data management tasks and for input/output of data to/from the database while the 'middleware' is the web server that acts as the dynamic glue between the front-end user form and the back-end database by providing the business application logic that both constructs queries to interrogate the database (based on user input) and then constructs in real-time a webpage of data results to be sent back to the web browser. A good diagram is essential for full marks.

Examiner's comments :

Part (c) was the poorest of the three sub-questions. While most candidates managed some type of reasonable 3-tier diagram, the distribution of presentation/logic/data (and the specific function of forms and databases) was either not addressed at all or poorly expressed. There was clearly some confusion between the ANSI-SPARC 3-tier DBMS model and the web-database 3-tier model.

Question 6

General Comments (Q5)

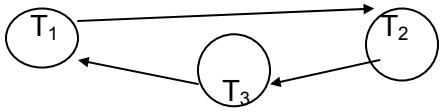
This was an unpopular question with 30% of the candidates attempting it. Of those, 30% achieved pass level marks.

Answer pointer part a :

(i) The answer needs to identify the conflicts between the transactions, either as shown in the diagram below, or by describing those conflicts. (3 marks)

T1	T2	T3
read(X)	read(X)	read(Y)
X := X – 3		Y := Y + 1
	read(Y)	
	X:=X+2	write(Y)
read(Z)	write(X)	read(Z)
Z:=Z + 1		
write(Z)		

Conflict Graph:



Since there is a cycle, the schedule is not serialisable

(i) The answer should describe any concurrency control mechanism. For example, the 2-phase locking protocol guarantees serialisability. Under the 2-phase locking protocol, all locking operations in a transaction must precede the first unlock operation in the transaction. Alternatively, use time stamps.

Examiner's Comments:

It was disappointing to see that almost all candidates struggled with this question. Only one used a conflict graph to illustrate their answer.

Answer pointer part b :

Pessimistic concurrency control protocols delay operations to ensure that conflict does not occur. This includes locking-based protocols where locking is used on the assumption that conflict will occur and therefore needs to be avoided.
Optimistic concurrency control protocols assume that conflicts are rare and allow transactions to proceed without imposing any delays (instead checking that serializability has not been broken at commit time). Timestamping is such protocol.

Examiner's Comments :

There were some good attempts at this part of the question

Answer pointer part c :

Timestamp: A unique identifier created by the DBMS that indicates the relative starting time of a transaction. (2 marks)

Timestamp methods for concurrency control are quite different from locking-based protocols:

Timestamping involves assigning timestamps to transactions, and controlling or checking access to data by timestamp order. No locks are involved, and therefore there can be no deadlock.

Locking methods generally prevent conflicts by making transactions wait. With timestamp methods, there is no waiting; transactions involved in conflict are simply rolled back and restarted.

Examiner's Comments:

Although the notion of timestamping is well known to most candidates, there is little understanding of how the timestamping protocol works.

Answer Pointer part d :

Log file contains before and after-images of updates to the database. Before images can be used to undo changes to the database; after-images can be used to redo changes.

Examiner's Comments: Most candidates have a good understanding of the use of log files in recovering databases.