THE BCS PROFESSIONAL EXAMINATIONS
BCS Level 5 Diploma in IT

October 2007

EXAMINERS' REPORT

Database Systems

General Comments

Some students still had difficulty in indicating on the front sheet which questions they had answered but generally there has been an improvement in this area. The majority of students answered the required number of questions, only a small minority answered fewer questions than the required amount.

Question 1

Discuss and explain all the relevant concepts and constructs that must be addressed and implemented by a database programmer or database administrator when satisfying the following database issues:

a) Security constraints (on users) **(15 marks)**
b) Integrity constraints (on data) **(10 marks)**

You may use your own SQL code examples and any appropriate diagrams to support your answer to a) and b).

Answer Pointers

Part a) required students to explain all relevant concepts and constructs for ensuring database security and integrity. Students need to show they understand the difference between security and integrity. The answer needs to look at areas such as authorized database access using user id and password; hiding sensitive data using views; managing system and database object privileges using grant/revoke and the use of roles to simplify privilege allocation. A discussion was needed on each of these areas followed by SQL code examples to illustrate how each area is supported.

Part b) was on the topic of integrity constraints and students needed to discuss entity integrity using primary key constraint; referential integrity using foreign key constraint and domain integrity using check constraints. Again the students needed to supply SQL code answers to illustrate this part of the answer.

Examiner's Guidance Notes

Most students were able to answer this question satisfactorily however some students omitted the SQL examples thus losing some marks. For part b) domain integrity proved to be a bit of a problem for some of the students but otherwise they provided good answers on entity and referential integrity.

Question 2

a) Using examples and diagrams where appropriate, define the following Relational Algebra Operators:
Project, Select, Union, Intersection, Join, Cartesian Product, Difference, Divide.
(16 marks)

b) Consider the following schema:

Employees (Empno, Ename, Dept, Job)
Skills (Skill_no, Empno)
Dept (Deptno, Deptname, No_Employees)

Write the relational algebra operators and steps required to compute the following sets:

i) List the names of the departments with more than 50 employees.
(**3 marks)**

ii) List the numbers and names of departments with at least one employee who has skill_no 5.
(6 marks)

Answer Pointers

Part a) Students should start off by defining each of the relational algebra operators and then provide suitable examples using sample data.
Select/Restrict – extract tuples which match a certain condition
Project – extract certain columns from a relation rather than displaying all of the columns in the final result.
Union – combining data from two relations which are union-compatible ie they have the same number of columns which are based on the same domains. Data which appears in either relation or both relations is given in the final result. So a union could not be carried out on a relation which has two columns and another relation which has 4 columns because they do not meet the union-compatibility rule.
Intersection – again the relations need to be compatible but only the data which appears in both relations is given in the resulting relation
Difference – eg for relation A – relation B the resulting relation would consist of data that appears in relation A but does not appear in relation B.
Join – The result of the natural join is the set of all combinations of tuples in two relations that are equal on their common attribute names.
Product - If the input relations have *N* and *M* rows, the result will contain *NM* rows.
Divide - The result consists of the restrictions of tuples in *A* to the attribute names unique to *A*, i.e., in the header of *A*but not in the header of *B*, for which it holds that all their combinations with tuples in *A* are present in *B*

Part b) students needed to demonstrate their practical knowledge of relational algebra by applying the relevant operators to achieve the two queries. The first query was simple and required the use of a select/restrict to extract those departments with more than 50 employees. Then a project was needed to just extract the deptname column for the result.
The second query was slightly more complex. It required a select/restrict to extract those employees who had skill 5. Then a join was needed with the employee relation to get the deptno. A further join was needed with the dept table because this contains the deptno and deptname columns. Finally a project was needed to extract the deptno and deptname columns for the result.

Examiner's Guidance Notes

This question also had a good pass rate. Students were able to give basic descriptions of most of the 8 relational operators. The divide relational operator proved to be difficult for some students. Most students supplied good examples to support their answer but those who didn't lost marks as a result. For part b) most students who attempted this section generally did quite well although a few did struggle with the second query.

Question 3

Suppose that a Housing Authority wants to use an Internet database application in its inventory of low-income housing units. Assume that the users of the application reside in a number of different government and non-profit organisations.

a) Sketch a possible three tiers system for supporting this Internet database application.
(10 marks)

b) Assuming that the data resides in an MS Access and that Windows 2000 is used on the Web Server, describe the functions of each of the tiers, you have produced in the previous question, and of the major components on each of the tiers.
(15 marks)

Answer Pointers

(a) The ANSI-SPARC three level architecture uses three levels of abstraction: external, conceptual and internal.

- The external level concentrates on modelling the users' views of the database without worrying how it the database is represented conceptually or physically.
[3 Marks]
- However, the conceptual level concentrates on modelling the community view of the database. It specifies the information content of the entire database, independent of storage consideration. It represents all the entities, their attributes and their relationship as well as the constraints on the data, and security and integrity information. It is mainly concerned about the overall logical view of the database.
[3 Marks]
- The internal level concentrates on the computer's view of the database. It is concerned about how data is represented, how records are sequenced, what indexes and pointers exist.
[3 Marks]

There is logical and physical data independency. The logical data independency is between the external and conceptual level while the physical exits between the conceptual and internal level.
[1 Marks]

(b) The two tiers architecture came to accommodate an increasingly decentralised business environment by providing a basic separation of tasks, which used to be in one place in the highly centralised business environment. The client (tier 1) is primarily responsible for the presentation of data to the server is primarily responsible for supplying data services to the client.

The need for enterprise scalability challenged this traditional two-tiers architecture. In the mid 1990s as application become more complex and potentially could be deployed to hundreds or thousands of end users. As a result the client side faces two problems; the need for considerable resources to run effectively as it becomes more and more 'fat' in addition to that it requires a considerable client-side administration overhead. To solve the above problems and the scalability of the enterprise, the three tiers or layers which can run on different platforms:

1. The user interface layer, which runs on the end user's computer,

2. The business logic and data processing layer, a middle tier which runs on a server called application server and
3. A DBMS, which stores the data required by the middle tier. This tier may run on a separate server called database server.
   As a result of this new architecture, the client is now responsible only for the application's user interface and some simple logic processing, and the core business logic now resides in its own layer.

[10 Marks]

The three tiers architecture maps quite naturally to the web environment, with the web browser acting as a thin client, and the web server acting as the application server. It also responds to the web environment where the number of user is very large and also platform independency, etc…

[5 Marks]

Examiners' Guidance Notes

The aim of the question is to assess the students' understanding the difference between the ANSI-SPARC three level architecture for relational database systems and the two/three tiers Client/Server architecture used for web database or e-commerce databases. Students' answers show that students do not really know the difference between the two architectures. So, only a small number of students have answered correctly both questions and showed that they differentiate between an ANSI-SPARC three level architecture and a two/three tiers Client/Server architecture. However, most of them have answered treated question (a) and (b) in a similar way i.e. as it is about two/three tiers Client/Server architecture instead of ANSI-SPARC architecture. The examiner suggests that candidates should 'read around' the two architectures and go through real example to appreciate the difference between the two architectures and their use.

Question 4

Consider the following database scenario:

"A part manufacturing company has one engineering department in Manchester and two manufacturing plants, one in London and one in Hong Kong. Each part type is produced at only one manufacturing plant. Currently, the company has one database located in the Engineering department in Manchester. Applications at the manufacturing plants access this database via a communication network for whatever data they need.

One of the relations in this centralised database system is the PART relation, where data about the manufactured parts are kept. The attributes of the PART relation are: the part's number (Part#), part's name (Name), part's manufacturing cost (Cost), the part's drawing number that specifies its design (Drawing#), the name of the plant where the part is manufactured (Plant), and the quantity manufactured up to now (Qty).

An instance of the **PART** relation is the following:

PART					
Part#	**Name**	**Cost**	**Drawing#**	**Plant**	**Qty**
p2	Widget	200	123-7	London	500
p7	Gizmo	600	501-9	Hong Kong	1000
p3	Thing	100	238-2	Hong Kong	2000
p1	Gadget	1000	310-0	Hong Kong	40
p8	Acme	150	400-6	London	3000

The company has decided to move to a distributed database system where each of the sites has its own database."

i) Propose a fragmentation design of the PART relation that reflects the distribution of the company's sites and their functionality.

(8 marks)

ii) Justify your proposed fragmentation design for the PART relation.

(5 marks)

iii) Write an SQL statement for each of the fragments obtained in your design.

(12 marks)

Answer Pointers

i)
There might be different ways of distributing the PART table. One strategy of fragmenting the PART table is as follows:

First, the relation PART is firstly fragmented vertically. The first fragment (Part#, Name, Drawing#) is about MANCHESTER engineering data, whereas the second fragment (Part#, Cost, Plant, Qty) contains London and Hong Kong manufacturing data. Data about the Name and Cost attributes could be allocated in any site; however, "drawing" and "Qty" should be allocated to Manchester (where the design is done) and to both London and Hong Kong respectively (where the manufactured parts are stored).

Manchester-PART

Part#	Name	Drawing#
p2	**Widget**	**123-7**
p7	**Gizmo**	**501-9**
p3	**Thing**	**238-2**
p1	**Gadget**	**310-0**
p8	**Acme**	**400-6**

London-HongKong-PART

Part#	Cost	Plant	Qty
p2	**200**	**London**	**500**
p7	**600**	**Hong Kong**	**1000**
p3	**100**	**Hong Kong**	**2000**
p1	**1000**	**Hong Kong**	**40**
p8	**150**	**London**	**3000**

The **London-HongKong-Part fragment** is then fragmented horizontally into two groups of tuples (one for each engineering site) each of which is stored at the corresponding engineering site (i.e. London and Hong Kong). So London-HongKong fragment (Part#, Cost, Plant, Qty) is horizontally fragmented to two groups of tuples. The ones where Plant = "London" will be stored in London, those where Plant = "Hong Kong" will be stored in Hong Kong. The resulting fragmentation is as shown below:

London-PART

Part#	Cost	Plant	Qty
p2	**200**	**London**	**500**
p8	**150**	**London**	**3000**

HongKong-PART

Part#	Cost	Plant	Qty
p7	**600**	**Hong Kong**	**1000**
p3	**100**	**Hong Kong**	**2000**
p1	**1000**	**Hong Kong**	**40**

To eliminate the repetition of the Plant attributes "London" in London-PART fragment and "Hong Kong" in HogKong-PART fragment, the London-PART and HongKong-PART could be represented as shown below:

London-PART

Part#	Cost	Qty

p2	**200**	**500**
p8	**150**	**3000**

HongKong-PART

Part#	Cost	Qty
p7	**600**	**1000**
p3	**100**	**2000**
p1	**1000**	**40**

ii)
It is proposed a mixed fragmentation of the PART relation. First a vertical fragmentation based on the type of the plants i.e. engineering or manufacturing. As a result Manchester fragments will hold data (Part#, Name, Drawing#) necessary to the engineering departments while London and Hong Kong fragments will hold information (Part#, Cost, Qty) on the parts built at the local plant. In the vertical fragmentation the attribute Part# is included in both the fragments so that the original relation can be reconstructed. A further horizontal fragmentation is performed to allow information related to each plant to be held in a local database i.e. London and Hong Kong databases.
iii)

SQL statement for Manchester-PART fragment:

Select Part, Name, drawing
From PART

SQL statement for London PART fragment:

Select Part, Cost, Qty
From PART
Where Plant=”London”

SQL statement for Hong Kong PART fragment
Select Part, Cost, Qty
From PART
Where Plant=”Hong Kong”

Examiners’ Guidance Notes

Few students have selected this question. Students should be more exposed distributed database concepts and practice enough to grasp the concepts and appreciate the benefit of distributed database systems. The examiner suggests that candidates should practice more on distributed database applications to be able to design database applications with advanced requirements which are not met by central database systems.

Question 5

a) Using specific examples from the Scenario (Appendix A) outline the main differences between a File-based approach to data processing with a Database-based approach to data processing.

(5 marks)

b) Produce a logical database design for the scenario described in Appendix A. Your design MUST include each of the following:

i) An Entity-Relationship (ER) diagram using the Entities shown in bold font.

ii) A list of Table definitions that includes column names, primary keys and foreign keys.

You must state:

i) The ER diagram notation used, using examples from your ER model.

ii) The method that was used to transform your ER model into Tables.

iii) Any assumptions that you have made in your modelling.

(20 marks)

Answer Pointers (a)

The main points are for 2 marks these being:
1. Example reference the data feeds – these would be discretised and stored as text files, similarly GPS data is encoded (NMEA format) and this would be stored in text files. Differences in approach are evident in processing and storage management:

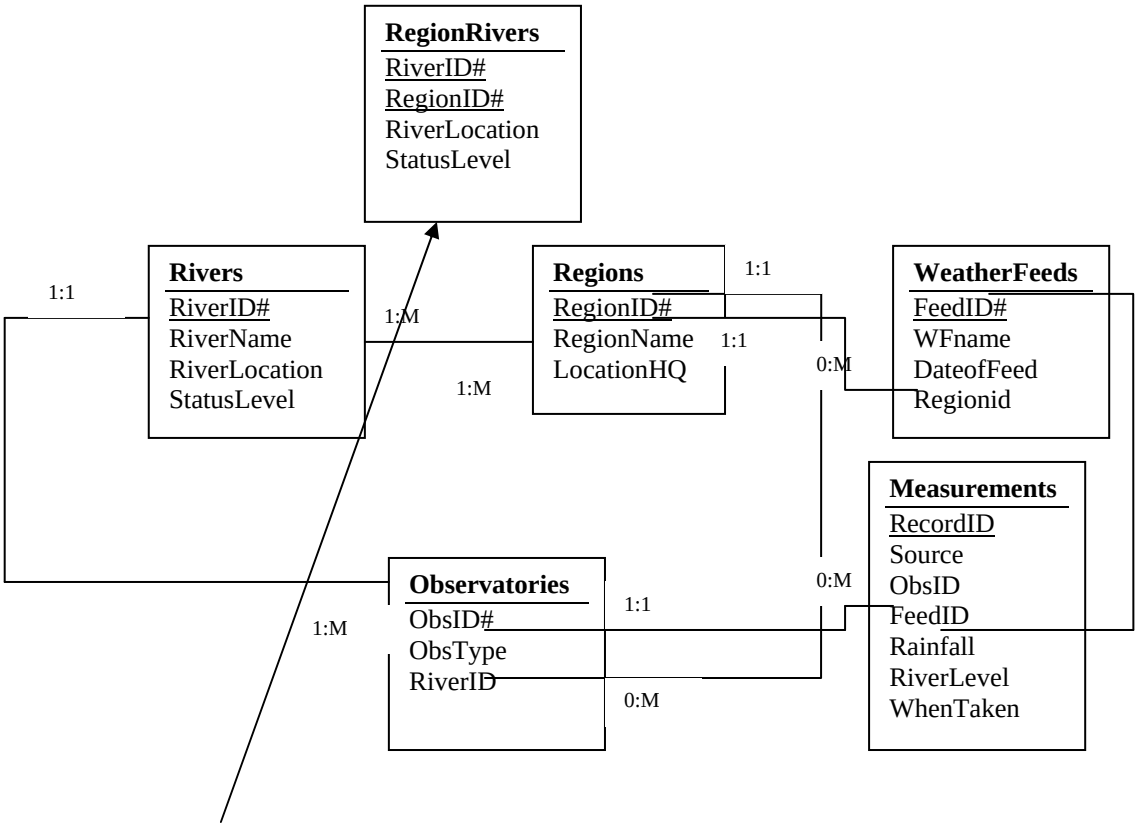
3 marks for each of the following points

- Files stored within and controlled by Operating system – DBMS it is not – it is controlled by layers of access ports.
- Files need special proprietary software for reading and writing usually low level whereas a DBMS abstracts the access to a relational view within SQL – a standard access port.
- Indexing not built in a file based approach would again be an operating system feature whereas DBMS has the best of both can index text files as separate linked entities or embedded in columns – means DBMS dictates the most efficient policy

Examiners Guidance Notes
Candidates' answers were generally encouraging best answers applied examples to the Case study.

Answer Pointers (b)
The ERD is supplied overleaf. The ERD is not too complex. The only modelling decisions to account for are the use of a linked entity and what weather feeds are related to in terms of other entities. The ERD is a logical model so the tables can be mapped using the algorithm supplied. This is given as many candidates do not seem to be aware of this technique. {aside the book by ER data analysis for database design covers this technique}

Data Model Answer part b)



NB Linked table reconciled
Notation used is UML example An Observatory records 0: many measurements, a measurement is associated with 1 and only 1 observatory.
A river is associated with NONE or MANY Regions. A region has ONE or MANY rivers flowing through it

Table Mappings achieved using the following rule
1:1 – 1:1 – Single Table form 2 entity types primary key chosen between 2 identifiers
0:1 – 1:1 – 2 tables with FK on 0:1 side posted to entity on 1:1 side
1:1 – 1:m – 2 tables as above (PK…FK1) PK=FK1 in linked table
1:M – 1:M - 3 tables – create a composite key from 2 PK of entity1 and entity2
TABLES
Are as defined following reconciling rules above containing the attributes as posted in the tables. Critical data type decisions include DECIMAL(5,2) for data values and CHECK constraints containing a regular expressions for matching url reference to data feeds

Examiners Guidance Notes

Candidates answers are improving as candidates seem to be better prepared on this regular type of question. There are still many weaknesses such as not reading the discourse and perhaps reading too much in the discourse. If there are any modelling decisions that contradict the discourse then marks cannot be awarded. There was one common mistake that arose – creating an unnecessary EA company entity. There is only one record so an entity is not justified. Table mappings where unclear in many cases made worse by the absence of a technique or even a commentary. This implies many candidates are not applying data modelling in real world situations.

Question 6

a) A travel agent has a late booking service where customers can book available seats on flights a day or so before the departure date. Sales are brisk as there are many opportunistic customers trying to get seats as discounted prices. Here's a telephone conversation between the ticket agent and a customer.

Customer: "I'd like two tickets to Honolulu for January 15."
Ticket agent: "OK, I have two seats on flight number 192 leaving at 10 a.m."

Customer: "What is the price?"
Ticket agent: "£900"

Customer: "Great, I'll take them."
Ticket agent: "The screen has refreshed and the price now is £920"

Customer: "No problem please book them."
Ticket agent: "Oops. I just tried to reserve them, and they've been sold by another agent. Nothing else seems to be available on that date."

i) Explain the nature of the problem that has occurred above.

(5 marks)

ii) Describe the mechanisms that a server-based DBMS would use to prevent the problem above.

(6 marks)

b) Explain the function of the following program:

```
CREATE TRIGGER TestTrig2 ON TestTab FOR UPDATE AS
SAVE TRANSACTION MyName
INSERT INTO TestAudit
 SELECT * FROM inserted
IF (@@error <> 0)
BEGIN
  ROLLBACK TRANSACTION MyName
END
```

(8 marks)

c) Describe the differences between the RECOVERY mechanism in b) above and the RECOVERY mechanism that occurs when a DBMS crashes or is suddenly shut down.

(6 marks)

Answer Pointers (a, i)

This is a standard database concurrency problem.
5 marks broken down as follows

3 marks for identifying and explaining the concept of concurrency control
1 mark for Dirty Read
1 mark for uncommitted dependency
2 marks for relating this to the problem defined and not using bookwork i.e. transaction integrity has defaulted due to same seat being booked at the same time. Price changed from previous transaction read.

Examiners Comments
Weak candidates did not spot the two situations and perhaps relied too much on bookwork resulting in low marks

Answer Pointers (a, ii)

The mechanism a database uses to prevent concurrent processes interfering with each other is called Locking ensuring the processes are serialisable and schedulable all of these techniques are applied using a 2Phase Locking (2PL) protocol
Candidates should explain the way that Dirty read and Uncommitted Dependency are prevented by locking: 3 marks, by 2PL:3 marks

Examiners Comments
Generally satisfied with most candidates answers weak students couldn't connect there answer to the scenario and relied too much on bookwork.

Answer Pointers (b):

For 3 marks identify the type of program – a UPDATE TRIGGER that fires in response to an event – the INSERT statement

For 3 marks identify the recovery mechanism built in and the meaning of @error.

For 2 marks an illustration of transaction integrity used to enforce the trigger

Examiners Guidance Notes

About 50% of candidates seem to be familiar with triggers. Of those that are not produced 30% with blank answers and 20% thought a trigger was a stored procedure and not an event activated database program. Again a bit disappointing to see this lack of exposure to real world database programming techniques.

Answer Pointers (c)

For 3 marks candidates must describe how the DBMS provides snapshots and transaction logs that record from the checkpoint of last known savepoint, whereas (for 3 marks) the trigger approach is a programmatic approach that has lower priority because of its 'local effect' – DBMS always gives precedence to global effects or events.

Examiners Guidance Notes

Very weak attempts in most cases. Only about 20% of candidates were reflective and knowledgeable enough to see the differences. It was disappointing to see that many candidates could not seem to connect 2 different problems covered in different chapters of most text books. Again seems to point to a lack of real world database exposure.