THE BCS PROFESSIONAL EXAMINATIONS
Diploma in IT

October 2006

EXAMINERS' REPORT

Database Systems

General Comments

Some students still had difficulty in indicating on the front sheet which questions they had answered. I have said this many times before, perhaps student attention needs to be drawn to this and a sentence is required on the front sheet which states 'Please circle the questions you have answered'. On this sitting, I'm pleased to say that nearly all students answered the required number of questions. Students must take care on producing legible answers with correct spelling and/or grammar.

Please note that candidates are expected to demonstrate ***practical application*** of their knowledge.

Question 1

1. *a)* Compare and contrast the file-based approach to data management with the database approach. In your answer you should discuss the advantages and disadvantages of each approach. **(9 marks)**

b) The following outline schema was developed for a customer company database for a large organisation manufacturing and wholesaling a wide range of chemicals for agricultural pest control and land fertilization:

Customerno, customer name, customer address, invoice address, delivery address, contact name, contact_tel_no, customer type, product category, delivery day, salesmanno, credit limit, credit terms, invoice date, discount, turnover this year, turnover last year, current balance.

Identify some potential users of this data and specify suitable external schemas for each. **(7 marks)**

c) Give a definition of data independence? Distinguish between logical and physical data independence. Illustrate your answer with an example. **(9 marks)**

Answer Pointers

a) File based approach:

- Structure of Data file dependent on Application program, changes to a file format will affect all programs using this file, leading to extra maintenance and programming effort Redundant and duplicated data spread across many files, as a result possible inconsistencies between two or more different data items
- Changes must now be propagated to multiple copies of the same data
- Distributed / replicated data means less security (no centralised control) or security is more difficult to manage.
- Duplicated data may lead to unnecessary and wasted storage
- Manipulating the file data requires technical (imperative) programming skills
- No support for a higher-level query language like SQL
  Possible advantage = ok for data which is not organised in a complex manner, no additional investment is required, ok for small org where only handful of users who have same access to the data.

DBMS approach:

- Centralised control via DBMS
- Greatly enhanced security
- Minimal duplication and redundancy of data, easier maintenance

- Data independence from application programs, easier to carry out changes to structure of data.
- Supports for high level (declarative) query language like SQL
- More intuitive and less 'low-level' than writing 3GL code
  Possible disadvantages – invest in purchasing software, training staff to learn how to use the dbms.

b) SALES - customer#, customer name, customer address**,** delivery address, credit limit, credit terms, discount
ACCOUNTS - invoice date, invoice address, current balance, turnover this year, turnover last year
WAREHOUSE - customer#, customer name, customer address, delivery address, contact name, contact tel no, delivery day

c) data independence – data is independent from the application program.

Logical data independence concerns the external level and conceptual schema. It allows changes to be made to the conceptual schema without necessarily affecting the external schema.

Physical data independence concerns the conceptual schema and internal schema. The conceptual schema is immune to changes in the internal level. Definitions along these lines together with examples of each type of data independence should be given.

Examiner's Comments
This question was answered by the majority of the students and in general the basic information provided in the answers to part a) was correct. Fewer students understood what was required for part b) and only a small minority managed to get high marks for this section. Part c) was answered well in most cases and students were able to provide an example of each type of data independence. To summarise, students showed a good understanding of this topic area.

Question 2

2. *a)* Security is of paramount concern when using multi-user systems. Briefly explain the following security issues that arise in a multi-user system:
i) authentication of users
ii) user privileges
iii) confidentiality of data **(6 marks)**

b) Describe the DBMS software facilities that are used to support the above security issues in a multi-user system. Give examples (using sql code) of user authentication, user privileges and confidentiality of data. **(12 marks)**

c) The Computing Faculty in a University holds information on students, assessments, lecturers, courses and modules in a centralised DBMS. Users of this database fall into the following groups: Administrators, students, Course Administration Staff, and Personnel. What role would the Database Administrator (DBA) play in this situation to ensure security and integrity of the data? **(7 marks)**

Answer Pointers
a) i) users have to be authenticated to prevent unauthorised access to the data in the dbms. Authentic users will have a user account and also know the password for their account.
ii) Once we have determined who the authorised users are, we need to set out access privileges for these users i.e. which data do they have access to, do they just have read access or have create and update privileges been granted to these users.
iii) Some data in the centralised database is of a sensitive nature egg salary. We do not want every user accessing 'sensitive' data. Only those users who need the confidential data to carry out their job will be given access to this data. We can use views to make

sure that users only get access to the data that they need to carry out their jobs. A user view is a window to the database. The view definition determines the size of the window i.e. how much data the user can access.

b) user authentication – can be done by database password or operating system account password. A 'Create User .. Identified by …' type example would be needed here.
User privileges – give examples which make use of the sql grant statement.
Confidentiality of data – view definition example using sql.

c) The DBA needs to create accounts for all of the users and set up password information (which users should change when they login).
The DBA would create views for each user/user group to help determine the data that each user/user group needs to have.
The DBA would need to assign system and object privileges to each user.
This can be done using roles. So for example we could define a role for the Admin staff. Then each admin person can be granted that role rather than granting many privileges to each user one by one.
The DBA can ensure integrity constraints are set up appropriately for entity integrity and referential integrity.
This list is not exhaustive and other points can also be added.

Examiner's Comments

Again this question was answered by fewer students compared to question 1 nevertheless it still proved to be a popular question. There was a mixed response to this question. Some students managed to give perfect textbook answers highlighting points above and beyond those listed in the answer pointers. For other students the area of User privileges and views was better understood than user authentication. In the latter vague answers were given without any use of clear examples. Students covered the most of the relevant points in part c) but their answers lacked structure and were not coherent enough.

Question 3

3. *a)* To address the data independence in database systems the ANSI-SPARC three-level architecture was proposed. Compare and contrast the three levels of this ANSI-SPARC architecture. **(10 marks)**

b) Compare and contrast the two-tier client server architecture for traditional database systems with the three-tier architecture. Why is the latter (i.e. the three tiers architecture) more appropriate for the Web database? **(15 marks)**

Answer Pointers

a) This question had some overlap with question 1. Students were required to describe each level in the ANSI-SPARC three-level architecture, namely external, conceptual and internal. 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 mark)

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 can be achieved. Other points along these lines were credited with marks where relevant. (5 marks)

Examiner's Comments
Approximately half of the students attempted this question. Most answered section a) better and were able to gain higher marks compared to section b. In some cases, students described the mapping between the levels but did not expand their answer to discuss data independence. One frustrating observation was that for some reason quite a few students thought the database was stored in the internal level.

For those who attempted section b, the majority understood the difference between the two tier and three tier architectures and why the three tier architecture is more suitable for the web environment. However, a group of students albeit a small amount, misunderstood this section and confused 3 tier architecture with ANSI-SPARC three-level architecture and did not respond correctly to why the three tiers architecture is needed. The examiner suggests that candidates should 'read around' topics and also go through real examples to appreciate the need for such three tiers architecture for e-commerce database applications.

Question 4

4. Consider the following scenario:

A small Bank has two branches, one in Bristol and one in Bath, and has its head-quarters in London. Currently, it has a centralised database in its headquarters where it keeps data about its customers. Local use consists of report generation for trend analysis. On the other hand applications at the two branches access this database via a communication network for whatever data they need. There is also a communication link between the two branches, which is currently used only when one of the main links to the London headquarters fail.

The only relation in this centralized database system is the Customer relation, where data about customer accounts are kept. The attributes of the Customer relation are, the account number (Acc_no), the customer's name (Cust_name), the branch where the account is kept (Branch) and its current balance (Balance). An instance of the Customer relation follows:

Customer Relation

Acc_no	Cust_name	Branch	Balance
200	Jones	Bath	1000
324	Smith	Bristol	250
153	Gray	Bristol	38
426	Dorman	Bath	796
500	Green	Bristol	168
683	Roy	Bath	1500
252	Elmore	Bath	330

Due to heavy network traffic the bank's service to its customers is suffering. The Bank is concerned and has asked you to investigate database distribution designs that will improve its service.

Propose THREE distribution designs one for each of the following requirements:

- The database should always be available to all sites and access to it should be fast even in the case of data communication link failures. **(7 marks)**
- There should be no redundancy in the allocation of data, i.e. only local data should be stored at a site. **(10 marks)**
- A reasonable compromise between requirements *i)* and *ii)*. You should also justify each proposal and outline its advantages and disadvantages. **(8 marks)**

Answer Pointers

(Propose three distribution designs one for each of the following requirements):

i) The database should always be available to all sites and access to it should be fast even in the case of data communication link failures.

ii) There should be no redundancy in the allocation of data, i.e. only local data should be stored at a site.

iii) A reasonable compromise between requirements i) and ii).

A reasonable set of proposals could be:

i) The Customer relation is replicated to all three sites. Advantages: Availability and fast access. Disadvantages: difficult to maintain, update propagation, a lot of storage is required.
(7 marks)

ii) The Customer relation is horizontally partitioned into two fragments, which are stored one at the Bath branch, and one at the Bristol one. The London headquarters do not store any data. Fragment Bath = **s** (Branch="Bath") and Fragment Bristol =**s** (Branch="Bristol"). Advantages: No redundancy of data, easier maintenance and security, efficient use of physical storage. Disadvantages: Data becomes unavailable is a site or links to it fail.
(10 marks)

iii) The relation is horizontally fragmented and allocated to the Bath and Bristol branches as above, but a copy of the whole relation is also allocated to the London Headquarters.
(8 marks)

Examiners' Comments

Few students have selected this question. It is apparent that students have not been well exposed to topic of distributed database concepts and did not practice enough to grasp the concepts. 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

5. Please refer to Appendix A at the back of this paper for this question.

a) Construct an ER diagram containing Entity Types, Relationship Types and Degrees to give a high level model of the SWIFT information system described in **Appendix A**. Your model should have no more than 10 Entity Types. **(10 marks)**

b) Extend your ER diagram by assigning the columns listed in the tables in **Figure A1** and **Figure A3** to the appropriate Entity/Relationship Types. Use your ER diagram to explain the interdependencies that exist between the data contained in **Figure A1** and **Figure A3**. **(7 marks)**

c) Explain with the aid of examples obtained from **Appendix A** the concept of ***Relational Integrity*** in a data model and explain how ***Relational Integrity*** is translated into SQL code. **(8 marks)**

STATE any assumptions made in your modelling.
STATE the diagram notation you have used in part *a)*.

Answer Pointer Part a)

The ER model has two significant Entity Types and these are dictated by the views implied in the data set supplied in other words restricted to SWIFTS view of data across traders and customers.

The core Entity Types being:
SUPPLIER| ORDERS (who they purchase from)
and **CUSTOMER ORDE|RS** (who they sell to).
Other Entity types (ETs) are needed to support these 2 core Entity Types and thus the modelling should be broken down in this way and this make the model easier to formulate. Consider the Relationship Types (RTs) names and degree 1 to Many or Many to many.

Identification of other non-core Entity and Relationship Types (skeleton entities is now possible. The additional ETs are highlighted in bold font below.

Customers -→ CustomerOrders (a 1:Many relationship)
Relational Type= Purchases Order relates to a unique customer.
Suppliers -→ SupplierOrders (a 1:Many relationship) RT=Has
SupplierOrders **-→ Products (M:N) RT – Has**
CustomerOrders **-→ Products (M:N) RT - Has**
Prices -→ Products (M:N) RT 'For'

The Price Entity Type models the historical change in prices per product over time (this is a M;N) relationship as well and can be decomposed as
PriceCode -→ OfferCode ←ProductCode

(Please note no extra marks were given for participation constraints)

The notation used must be specified – in this case:
→ is a relationship between 2 Entity Types and must be qualified as a 1 to Many or Many to Many (M:N)

Part b)
First of all 1:M relations must be broken down the M:N relationships and this means new ETs will be introduced thus:

SupplierProductOrderLine and **CustomerProductOrderLine**

These new ETs require identifiers such as SORDERNo and CORDERNo and attributes for the price and quantity for each product on the order.
Attributes are assigned by determining the functional dependencies between the identifiers and known attributes in the discourse. Thus:

COrders(CorderNo,CustID,OrderDate)
SOrders(SOrderNo,SupplierID,OrderDate)
Customer(CustID etc ….
Suppliers(SupplierID etc …
COrderLine(CorderNo,ProductCode,QtyOnOrder,PriceCode)
Prices(PriceCode#,Price,DateValid
Offers(Offer#,AirlineID,Flight,NoTickets
PriceCodes(PriceCode#,Offer#,Price,Valid_date)

Notation # = PK

Part c)
Referential integrity is an important concept that can be applied to any relational model to achieve integrity during updates to data referenced in two or more tables which have a matching set of values.
For example employee and worksFor.
An employee in WorksFor has an employeeID and a department and it is the employeeID that must link to the employeeID in Employee otherwise that employee cannot exist. Therefore Primary Keys are good candidates to be referenced in other tables supporting integrity checks such as the SQL 'Exists and 'IN' operators.

SQL can define referential integrity and this is built into the SQL standard (v92 or v99).
For example there is an explicit FOREIGN KEY REFERENCES statement build into CREATE TABLE or ADD CONSTRAINT.
Here is a typical example that candidates should produce

```
CREATE TABLE order_part
    (order_nmbr      int,
    part_nmbr       int
       FOREIGN KEY REFERENCES part_sample(part_number,
    qty_ordered int)
```

The example should show how a foreign key in one table points to a candidate key in another table. Foreign keys prevent actions that would leave rows with foreign key values when there are no candidate keys with that value. In the following sample, the order_part table establishes a foreign key referencing the part_sample table.

Examiner's Comments
Although part b) was dependent on part a) many candidates lost marks by combining their answers into one model and therefore not showing their working out in going from the schematic model of part a) to the detailed model of part b)

Part b) should have covered reconciling M:N relationships and applying functional dependency theory to ensure some degree of (informal) normalisation. The examiner could not see this stage in the modelling process from some candidates, which is why there were two parts to the question. Marks were lost if there was no notation defined for ER model symbols although candidates who simply drew tables rather than use ERD) were not penalised as this question focused on relational modelling and the delivery of a logical schema.

Broadly 4*2 marks were given for each correct Entity – Relationship pairing (with assumptions if applicable)
Good candidates managed to reconcile Many to Many relationships but there were not any candidates that considered the possibility of a connection trap as a means of reconciling 3 way relationships. (An extra 2 marks if this was considered).

Part c) was generally answered well but would have like examples pertinent to the model produced in part b).

4 marks for explanation (1/2 good examples expected)
4 marks for SQL code sample

Question 6

6. *a)* Outline the differences between a static SQL query and a dynamic SQL query. **(6 marks)**

b) Describe with the aid of diagrams the function of database middleware when used to support interaction between a web browser and a database server. **(7 marks)**

c) JDBC is database middleware supported by Java. Explain the function of JDBC in database access using the sample Java code fragment given below in **Figure 6.1**. **(6 marks)**

d) Write a fragment of JDBC code or pseudo-code to show how a database connection is made using Java (or equivalent application programming language). Briefly explain how your code works. **(6 marks)**

```
{
String sqlStmt = "SELECT * FROM CUSTOMER WHERE ";

if(txtCity.Text.length() > 0)
   sqlStmt += " Address2 = '" + txtCity.Text + "' "; // London ;
 }
```

Fig 6.1 Java/JDBC code extract for use in Question 6

Answer Pointer Part a)

A static SQL query is where the SQL string formulating a request for a DB server is already known prior to execution. Most SQL code is of this nature as users submit SQL code on the fly or predefined in stored procedures.

Dynamic SQL query exists where a user (perhaps using a web client or similar) formulates the SQL query progressively using variable transitional steps for example an initial SELECT defines columns and tables then the where is a condition that is provided later. (3 marks for each)

Answer Pointer Part b)

Database middleware performs the function of connecting clients (egg web browsers on PC terminals) to a data source (maybe a different resource/ different server PC). The middleware supplies driver details so that the client can pass data to the database (SQL strings for example) ands connection properties (such as passwords etc). Therefore some brief understanding of ODBC/ OLEDB data providers should be part of the answer although at this level candidates are not expected to know the inner workings of middleware software but should be familiar with the 'black box' approach to middleware components (i.e. what do these components connect to etc). For this reason a diagram showing a 3 tier client server network with components such as client-server framework (via HTTP) ODBC or JDBC or bridged ODBC-JDBC for Microsoft-Sun connectivity (i.e. PC client with Oracle DB server)

Answer Pointer Part c)

This sample uses a dynamic SQL approach to pass a SQL string to a class called String which passes this to the JDBC driver for interpreting by JDBC middleware. The code provides some client side validation that would be costly and difficult to manage at the dB server side. So that address details matching the user input with the structure of the address field in the db is a particularly common task to prevent errors and capture these errors at the right time in the correct context.

Answer Pointer Part d)
A fragment of JDBC code or pseudo-code similar to the one below would be sufficient to show how a database connection is made using Java to a MySQL open source database server thus:

```
try {

Class.forName("com.mysql.jdbc.Driver").newInstance();
        con =
DriverManager.getConnection("jdbc:mysql:///test",
        "root", "secret");
}
```

Explanation would follow along these lines:
The above Java program tries to connect to "test" database installed on a MySQL server. "test" database is installed on a MySQL server by default so this is the database it connects to. The TRY is part of an exception handler that will detect whether the server is not running or it has been passed the wrong userid and password.

Examiner's Comments
This was one of those questions that required specific knowledge of the programming techniques almost certainly derived from practical experience of setting up web sites and programming database connectivity. Therefore this was not a popular question but of those candidates that tackled it most managed to get good marks and candidates utilised the flexibility of using other (non-Java) techniques that they had practised such as ASP.NET.

Appendix A: SWIFT (Shopping With Internet Futures Technology)

A1: INTRODUCTION

SWIFT is a company that sells a range of electronic products to its customers over the internet. Customers purchase products using an on-line order entry system called a 'shopping cart'. A collective has been set up involving SWIFT and a group of suppliers who supply SWIFT with the products they need to meet customer orders. This means SWIFT can negotiate discounted prices for specific products purchased from the supplier group. Each product has an initial price (discounted price) that decreases as the total quantity of products that SWIFT sell to customers increases. SWIFT also benefits from having a higher priority for the supply of products from the supplier group in the collective than other internet shopping companies. SWIFT make a profit on each product it sells as the discounted price includes a small commission that SWIFT receive on sales of that product. The time that a product can be bought by customers at the discounted price is called the offer period.

SWIFT raises supplier orders for products that customers have ordered from them.

SWIFT holds the products in a warehouse before dispatching them to customers. Suppliers can supply SWIFT with products on demand or 'just in time'.

Figure A1 (below) shows a sample customer order and contains data that was captured today (assume the date is 3rd December 2005). **Figure A2** shows those customers who have made orders.

Figure A1: TBL_CustomerOrders

COrderNo	CustID	ProductCode	Quantity	OrderDate	DiscountedPrice	ClosingPrice	ClosingDate
1962	343371	BG_8971	1	12/Nov/05	231.99	219.56	01/Dec/05
1962	343371	GTX_281	20	12/Nov/05	2.67	2.59	14/Nov/05
1962	343371	TL121_281	1	12/Nov/05	2040.00	2033.59	16/Nov/05
1963	034933	GTX_281	15	11/Nov/05	2.69	2.59	14/Nov/05
1964	984311	PD0045	5	12/Nov/05	2.67	2.59	16/Nov/05
1964	984311	JDYE_6	1000	30/Nov/05	25.99	22.37	30/Nov/05
1965	343371	TL121_800	1	02/Dec/05	2035.99	1994.72	03/Dec/05
1966	953534	GTX_281	1	03/Dec/05	2.57	NULL	NULL

Figure A2: TBL_Customers

CustomerID	Member	CustomerName	Address1	Address2
343371	Y	Andrews	123 Abel Ave	London
034933	Y	Ling Wing	6 Princes Street	Hong Kong
984311	N	Hutton	564 Holly Road	Manchester
953534	N	Rivers	80 Grange Way	Glasgow

SWIFT do not hold surplus stock in the Warehouse. **Figure A3** (below) shows an example of a supplier order.

Figure A3 TBL_SupplierOrders

SOrderNo	SupplierID	Product Code	Product Price	Quantity OnOrder	OrderDate	DeliveryDate
1004	B&G	BG_8971	219.56	10	01/Dec/05	02/Dec/05
1005	B&G	GTX_281	2.59	9	14/Nov/05	17/Nov/05
1006	B&G	TL121_2	2033.59	100	17/Nov/05	19/Nov/05
1007	JH Price	GTX_281	229.99	26	16/Nov/05	17/Nov/05
1006	B&G	PD0045	2.59	26	17/Nov/05	20/Nov/05
1008	JH Price	JDYE_6	22.37	900	30/Nov/05	NULL
1009	B&G	JDYE_6	22.37	100	30/Nov/05	NULL