THE BCS PROFESSIONAL EXAMINATION
Diploma

April 2003

EXAMINERS' REPORT

Systems Design

Question 1

a) Briefly explain the design issues that should be considered during the physical design of a Relational Database for EACH of the following:

(i)	***data types***	***(4 marks)***
(ii)	***derived (calculated) attributes***	***(4 marks)***
(iii)	***referential integrity***	***(4 marks)***
(iv)	***null values***	***(4 marks)***

b) When designing the physical tables it may be necessary to denormalize the given normalized tables in order to optimise for efficient data processing.

(i)	***Explain with an example what is meant by the term denormalization.***	***(5 marks)***
(ii)	***State two disadvantages of denormalizing tables.***	***(4 marks)***

Answer pointers:

(a) Physical Design decisions

(i) choosing data types

Candidates should discuss such things as: choice of data types that minimise storage space, can represent all possible field values, and allows the appropriate manipulation of the data, allows for business growth and can be validated. Any relevant examples accepted.

(ii) dealing with derived (calculated) attributes

Explain what a derived attribute is – mathematically determined by value of other attributes; if stored it uses more storage space but quicker to retrieve – must be updated when a change made to value of other attributes; if not stored but calculated slower to access but always accurate at that moment in time; required speed of access, therefore, important. E.g. How to deal with Total-Order-Cost in ORDER table.

(iii) controlling referential integrity

Referential integrity – value of attribute in one relation must have matching value in another (PK/FK). E.g. may not want to delete a customer record when there are orders outstanding for them. May control this by cascading deletes or restricting deletes. Updates need to be similarly controlled.

(iv) dealing with null values

Null values are 'missing' or 'unknown'and are not to be confused with a 0 value. Some values cannot be allowed to be null (PKs & FKs must have a value to allow joins – otherwise get Cartesian products). Null values may also affect the results of calculations on certain fields. Any relevant examples accepted.

4 x 4 marks for explanation and suitable example
total 16 marks

b) i) Splitting/combining normalised relations in order to optimise performance of the database. Other suitable explanations accepted.

ii)

- Can increase the chance of errors & inconsistencies that normalisation avoided
- Improves the performance of some queries at the expense of others

Any other relevant disadvantages accepted.
2 x 2 marks (total 4 marks)

Total for Q4 = 25 marks

Examiner's Comments

For part (a) of this question most candidates could describe each of the concepts (i) to (iv) but the weaker candidates did not explain the design issues involved for the concepts during the physical design of a RDBMS. For example most stated the various data types available (character, integer, date etc.) but failed to explain the significance of its choice from a design point of view – e.g. to represent all possible field values, allow the appropriate manipulation of the data, allow for business growth and enable validation. Several candidates misunderstood derived (calculated) attributes and described inheritance from the Object-Oriented model. In this case the attributes are inherited not derived. The word calculated was included in parenthesis in order to avoid misunderstanding.

In part (b) some candidates confused de-normalization with un-normalized attributes, and consequently described the process of normalization. The important issues in this part of the question were to recognize that de-normalization is carried out in order to optimize performance of common queries made on the database, and that the 'cost' of this is losing some of the advantages of a fully normalized database.

Question 2
Explain how each of the following would be used in Systems Design. Support your answer with an annotated diagram in each case:

(i)	***class diagram***	***(8 marks)***
(ii)	***sequence diagram***	***(8 marks)***
(iii)	***collaboration diagram***	***(9 marks)***

Answer pointers:
Candidate's explanation should include basic overview of what each model depicts, e.g.

(i) Class Diagram

- class diagram is a static view of the system
- shows all classes to support use case functionality
- class diagrams support the view of data as attributes & processes as operations.

(ii) Sequence Diagram

- dynamic view of the system
- shows interactions between objects in a time sequence

- Passage of time represented in sequence diagrams linked with processes (message passing)

(iii) Collaboration Diagram

- dynamic view of the system
- shows interactions (collaboration) between objects in a context
- on collaboration diagrams the sequence of messages is indicated by numbering and is not as clear.
- a collaboration diagram may be drawn for each route through a use case.
- collaboration diagrams use icons and a similar layout to class diagrams, and, therefore, are useful in checking the structure of the class diagram (particularly associations)

8 + 8 + 9 marks for suitable explanations & annotated diagrams
Total for Q5 = 25 marks

Examiner's Comments

This question was generally answered quite well, with many candidates providing clear and appropriate annotated diagrams to illustrate their explanations. Most of the candidates that scored poorly on this question did not know what a Collaboration Diagram was, nor what it was used for in systems design. Consequently, they lost 9 of the available 25 marks straight away.

Several candidates also demonstrated common misunderstandings. For example the Class Diagram is a static model and shows associations among the classes. These associations show that there will be 'some' message passing between classes but they do not show actual object interactions. These are seen as message passing in Sequence Diagrams.

Question 3

Identify and explain typical activities that might be undertaken at the design stage of an Information System. Include in your answer a consideration of both UML and Structured Design activities. (25 marks)

Answer Pointers:

Candidates should be able to discuss such activities as:

- identification of sub-systems & major components (packages)
- identification of concurrent processes
- allocation of sub-systems to processors
- selection of a data management strategy
- choice of strategy and standards for HCI
- identification of reusable components & class libraries
- identification of the control aspects of the system
- mechanism for error handling and fault correction

Candidates are not expected to cover all of the above points but to discuss a range of them and to describe how the various models are developed in order to support that activity. For example, in the structured approach to design structure charts are drawn in order to identify the software components needing to be developed; and for an object-

oriented approach UML models such as the package diagram and component diagram can be used.

Marks will be awarded for the range covered and quality of explanation.
Total for Q6 = 25 marks

Examiner's Comments

This question is asking candidates to identify typical design activities as outlined above. Generally marks were lost for this question by candidates failing to do so. Several candidates merely described models without saying why they were being developed, i.e. explicitly linking them to the above activities and demonstrating how the model helps that particular activity.

The weaker candidates described the whole of the Systems Development Life Cycle (SDLC), of which only the relevant points made concerning design could attract any marks. Clearly there was a lot of wasted effort here when the question plainly states that the design stage should be considered. Similarly some candidates described the object-oriented model in a very generalized way (i.e. terms of classes, inheritance, aggregation etc.) without dealing specifically with design issues. Marks were given for any relevant points made even where the candidates overall approach to the question was flawed.

Question 4

(a) Modern user interface technology enables a wide range of graphical and multi-media features to be incorporated into an interface. Outline how these features can help make the interface more usable. (5 marks)

(b) An office information system is to be used in a large open-plan office to help staff respond to customer phone calls and letters. Describe the advantages and disadvantages of using the following in such a situation:

- ***(i) Colour (4 marks)***
- ***(ii) Sound (4 marks)***
- ***(iii) Animation (4 marks)***
- ***(iv) Voice input (4 marks)***
- ***(v) Touch input (4 marks)***

Answer Pointers

a) All these features can be used to help reinforce metaphors, aid learning and remembering, and make the interface more interesting. Consistent use of interface features between applications can be helpful (eg IBM CUA). Simple examples like mouse-over highlighting and help, standard use of keyboard shortcuts (Alt F4), mechanics of drag and drop etc. (But anything too new or clever can be distracting).

Marks: up to 2 for points mentioned, up to 3 for saying how they help – max 5 marks

b)

Colour: Provides an easily recognised categorisation, e.g. fields changed, or in error; distinguishing between subsystems, also breaks up monotony of screen; Problems include colour-blindness, cultural differences in colour perception, e.g. colours denoting warning or danger.

Sound: This means simple sounds like beeps and clicks, e.g.Windows sounds for Start-up/close down, mail arriving etc. Can be useful for alerts/warnings; But can be obtrusive, especially in shared environment. Can also be irritating to user – e.g. frequent ‘clicking’

Animation: Can reinforce metaphors, e.g. depicting copying, moving, deleting, printing;
Can also be useful for 'how to do' help screens, e.g. MS Office Assistant,
but can also be distracting or irritating.
Voice input: May be good for hands-free use, note taking, or Word Processing say.
Probably bad in office environment, as with sound output.
Touch input: Simple interface, e.g. in public place, when typing is difficult,
or when keyboard/mouse inappropriate.
Limited usefulness, so probably not much use in this example.

Marks: one for each convincing point up to 4 per feature: 4 x 5 = 20 marks

Examiner's Guidance Notes
The better candidates showed some understanding of usability issues, and were able to quote simple examples to reinforce their answers. They were able to imagine using an IT system while working in an open-plan office environment, and identify the interface features that could help or hinder such work.

Question 5
Describe each of the following and give an example of how each may be used in a web based implementation:

(a) HTML	***(5 marks)***
(b) Java	***(5 marks)***
(c) JavaScript	***(5 marks)***
(d) SQL	***(5 marks)***
(e) A CGI scripting language	***(5 marks)***

Answer Pointers
- HTML – Hypertext Markup Language; sent from server to client; interpreted by web browser; describes page layout, colour, font etc., graphics, hyperlinks; only caters for static items.
- Java – full programming language, used either server or client side – Java beans, e.g. generating graphics and animation
- JavaScript – embed in HTML, adds simple dynamic/interactive capabilities to web page; e.g. basic validation, making items visible/invisible.
- SQL – Structured Query Language; allows database tables to be defined, updated and queried; e.g. links server code written in programming language to database.
- CGI script – Common Gateway Interface scripting language, eg VB, Perl.
  full programming language, runs on server, possibly interpreted like ASP,
  Can access database, validate logons etc., and generate tailored HTML to send back to the client.

Marks: 4 marks for each description and 1 for an example

Examiner's Comments
Most candidates knew about HTML and it's role, and also SQL. Most knew a bit about Java and its history, but not as many knew the role of Javascript, and relatively few understood about CGI scripts, most resorting to guessing (unsuccessfully). The better candidates were able to give examples of use, and show how some of the components link together.

Question 6

The illustration below shows a paper form for claiming expenses. The amount claimed for car travel is calculated by multiplying the miles travelled by the mileage rate. Taxable expenses are entered, and the tax paid is calculated by multiplying the taxable expenses by the current tax rate.

EXPENSE CLAIM

Name
Personnel No
Date of claim
Car Reg. No
Mileage Rate

Project

Method of payment: Cash
Cheque
Bank Transfer

Date	Description	Use of own car			Purchases	
		Miles	Claimed	Untaxable	Taxable	Tax
	Totals:					

Total amount claimed:

Total miles this year

From last claim:
This claim:
Total:

Signed (Employee) _____

Authorised (Manager) _____

(a) Draw the screens you would need to automate this system, and show how they are linked.
(b) Explain and justify any changes you have made compared with the paper- based system. (7 marks)
(c) List FIVE validations your system could perform. (5 marks)
(d) List FIVE advantages of your screen based system compared to the paper one (5 marks)

Answer Pointers

(a) There are many possibilities here, so marks were awarded for any sensible arrangement. Up to max 5 marks for main screen, 3 for additional screens or objects. Main form could be largely the same as shown, with a variety of protected and unprotected fields, drop-down lists, radio buttons, command buttons,

menu items etc., with the user entering figures as appropriate.
Some form of login/password may be needed at the start (although the system could rely on network login and obtain personal details using login id)
Some sort of control screen could be used to create, select, print and send for authorisation.
Maintenance form(s) for changing mileage and tax rates, adding users etc.
Some way of generating reports.

(b) Up to 3 points for mentioning changes, 4 more for justifying.
Signatures would not work! (unless last phase is to print and get signed)
Employees could not attach receipts!
It would not be possible to annotate comments unless explicitly provided.
Someone needs to keep mileage/tax rates, project lists etc. up to date.
What if employee didn't know if a particular item was taxable?

(c) One mark for each of 5 sensible advantages, e.g.
Numbers entered valid and realistic.
Dates valid and sensible.
Car reg. no. in valid form.
Personnel number is in database (or could be derived from network login username).
Project can be chosen from list of those applicable to employee
A method of payment has been selected, including bank details if appropriate.

(d) One mark for each of 5 sensible suggestions, e.g.
Calculating mileage, cumulative mileage and tax
Calculating totals
Could allow more lines of claims.
Entering user's name and personnel number automatically,
possibly car reg. no. as well
Automatically sending form to manager for authorisation, and to accounts for payment.
Part of a larger system, combining all expense payments, producing management reports etc.

Examiner's Guidance Notes
Most answers to this started well, but with a wide variety of approaches. Some took the form and implemented it unchanged, whereas some split it into many smaller forms.
The better answers showed the use of different screen objects within the form(s).
Explaining and justifying the changes proved rather harder, and some answers were very vague.
Validation also proved a challenge. Some answers gave very thorough validation suggestions, whereas others were extremely basic.
Many of the advantages given for the IT solution were largely wishful thinking, assuming anything computerised was necessarily better, although some were able to show real advantages. Not many candidates understood the limitations of an IT solution (eg inflexibility), but this was not specifically asked for.