BCS HIGHER EDUCATION QUALIFICATIONS

Level 5 Diploma in IT
September 2011

EXAMINERS' REPORT

DATABASE SYSTEMS

General

Table 1 shows the statistics based on the questions answered in questions Q1 and Q2. About 94.95% candidates attempted Q1 and 87.98% attempted Q2. Pass rate is 82.74 for Q1 and 66.21 for Q2. For standard deviation results, Q2 is higher than Q1. Thus it may conclude that these two questions are fundamental in database technology and candidates were aware about this importance.

Overall, from the evidence of marking results, the data modelling is done better, but the normalization is challenging to the candidates. However, compared with previous exams, the results in this examination have generally been improved. It follows that candidates are expected to improve the understanding in this area for BMS in their ability as the appreciation of these fundamental knowledge is important to the study in DBS.

TABLE 1: stats for all questions answered

	Q1	Q2	Q3	Q4	Q5	Q6	Total
Examiners (initials)							
Number attempted	395	366	259	328	95	81	
% Attempted	94.95%	87.98%	62.26%	78.85%	22.84%	19.47%	
Number Accepted	394	364	256	322	91	79	
% Accepted	94.71%	87.50%	61.54%	77.40%	21.88%	18.99%	
Number Passed	326	241	143	144	30	23	261
% Passed	82.74%	66.21%	55.86%	44.72%	32.97%	29.11%	62.74%
Max Mark	22	23	0	24	23	19	83
Min Mark	0	0	0	1	0	0	0
Average Mark	13.02	12.62	11.20	9.33	7.35	7.18	40.35
Standard Deviation	3.85	5.97	6.74	5.05	5.26	4.78	16.55

Question 1

The normalised relations (tables) below represent a bookstore database system.
Normalised relations:

Publisher	(PublisherID, PublisherName, City)
Book	(BookID, Title, Price, PublishedDate, PublisherID)
Author	(AuthorID, AuthorName)
Writing	(BookID, AuthorID)
Inventory	(BookID, BranchID, OnHand)
Branch	(BranchID, BranchName, BranchAddress)
Staff	(StaffID, StaffName, PhoneNo, BranchID)

a) Identify the foreign keys and list them in each relation; list the name of the table it is in and the name of the table it references.

[10 marks]

b) Use the information you have derived above to convert the normalized relations into an Entity-Relationship model. Label each relationship with a meaningful name. State any assumptions you make.

[10 Marks]

c) For each relationship in your models explain the choice of optionality in each direction.

[5 Marks]

INDICATIVE SOLUTIONS

a) Identify the foreign keys and list them in each relation; list the name of the table it is in and the name of the table it references.

[10 marks]

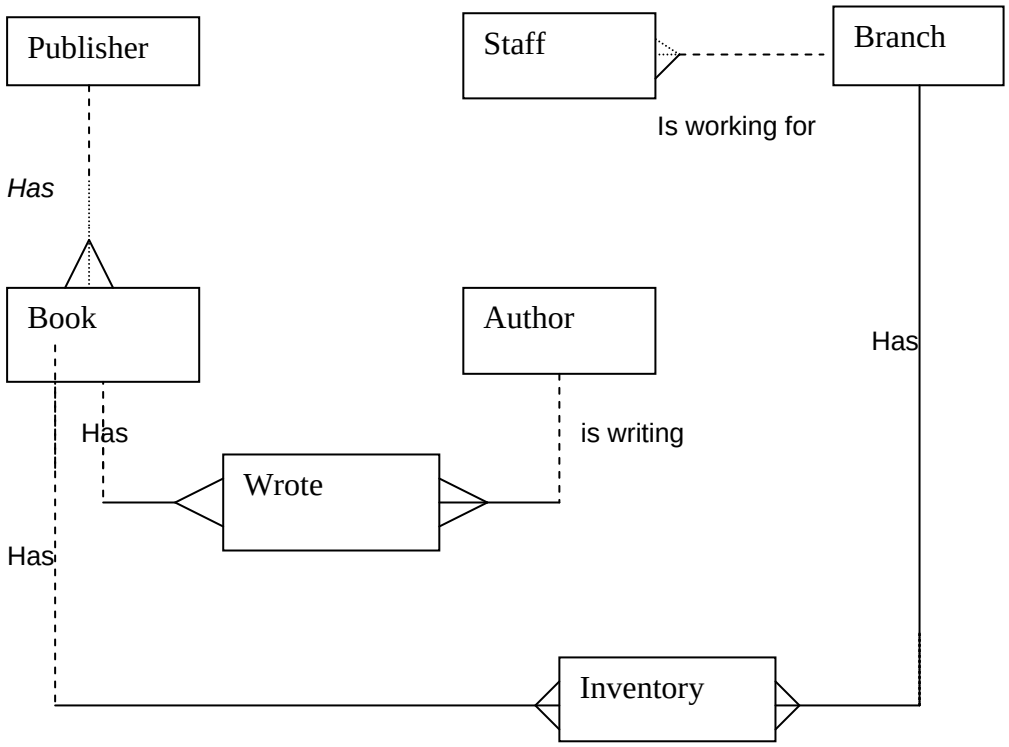
Table Name	FK	Referenced Table
Publisher	No FK	
Book	Published ID	Publisher
Author	No FK	
Writing	Book ID, AuthorID	Book, Author
Inventory	BookID, Branch ID	Book, Branch
Branch	No FK	
Staff	Branch ID	Branch

Candidates need to provide some elaborations for the answers above. For example, the tables without FK are Publisher, Author and Branch. Tables with FKs are Book, writing, Inventory and Staff, etc.

b) Use the information you have derived above to convert the normalized relations into an Entity-Relationship model. Label each relationship with a meaningful name. State any assumptions you make.

[10 Marks]

Dash lines are for foreign keys



Candidates need to elaborate the solution and assumptions they made.

c) For <u>each</u> relationship in your models explain the choice of optionality in <u>each direction.</u>
[5 marks]

The dash lines mean the relationship is optional.
The solid lines mean the relationship is mandatory.

Candidates need to elaborate the answers and assumptions they made.

Examiner's comments
This is a very popular question as about 402 candidates answered the question with an attempt rate of 94.95%, average mark 13.02/25, standard deviation 3.85 which is lower than Q2. About 82.74 % passed the question, which is the highest among the questions. Above statistic number demonstrates that candidates are interested in answering this type of knowledge in the subject area. The weakness for this question is that most candidates could not answer the section c properly or correctly, indicating the area needs further improvement.

Question 2

Concerning a database design, helpful information can be obtained from an existing flat file. The following is an invoice from a company named Summer Distributors. Determine the tables and attributes that would be required from this document.

a) Explain what is the normalization and why needs to do normalization in DBMS.
[5 marks]

b) Represent <u>each</u> attribute as an unnormalised relation and identify the candidate keys and primary key by underlining it.
[10 marks]

c) Normalize each of the unnormalized relations in (b) above, showing the results of each stage of the normalization. State any assumptions you make.

[10 marks]

BONo	PubCode	PubName	PubAdd	PubDate	ISBNno	Desc	Qty	Price
B001	P120	PublisherA	……	3/2/11	012345678-9	Fiction	3	20.00
					012345678-10	Fashion	6	25.00
B002	P121	PublisherB	……	3/2/11	012345678-11	Sports	3	30.00
etc.								

The attributes used in the form are:
BONo stands for book order number
PubCode stands for publisher code
PubName stands for publisher name
PubAdd stands for publisher address
PubDate stands for publishing date
ISBNno stands for book ISBN number
Desc stands for book description
Qty stands for quantity

Price stands for price of the book(s)

Answers and marking schemes

a) What is normalization?
[5 marks]

Normalization is a process to organize the data in an effective way, e.g. remove redundancy, improve manipulation operations, such as insertion, deletion, updating and so on for relational database management systems.

b) List each attribute as an unnormalised relations
Identify the primary key by underlining it
[5 marks]

Purchase_order – 0 (<u>BONo</u>, PubCode, PubName, PubAdd, PubDate, ISBOno, Desc, Qty, Price)

c) Normalization:

INF	[4 marks]
2NF	[4 marks]
3NF	[4 marks]
Assumption stated	[3 marks]

INF
Remove repeating group
Purchase_order-1 (BONo, Pubcode, PubName, PudAdd, PubDate)
PartsOnOrder-1 (BONo, ISBNno, Desc, Qty, Price)

2NF
Remove partial dependence

Purchase_order-2 (BONo, Pubcode, PubDate)
Publisher-2 (PubCode, PubName, PubAdd)
Book_order-2 (BONo, ISBNno, Qty)
Book-2 (ISBNno, Desc, Price)

3NF
Remove indirect dependence, above relationship is already in 3NF

Purchase_order-3 (BONo, Pubcode, PubDate)
Publisher-3 (PubCode, PubName, PubAdd)
Book_order-3 (BONo, ISBNno, Qty)
Book-3 (ISBNno, Desc, Price)

Examiner's comments
This is also a very popular question as about 368 candidates answered the question with an attempt rate of 87.98%, average mark 12.62/25, standard deviation 5.97 which is higher than Q1. About 6.21% passed the question. Above statistic number demonstrates that candidates are interested in answering this type of knowledge in the subject area, although the topic of normalization is always challenging in the subject area in both theory and practical level.

Question 3

Examiners Comments

This question was attempted by 62% of candidates and was a reasonably popular question. The pass rate of 56% was close to the overall average pass rate. It was obvious that those who had prepared for this type of question were able to score heavily, leading to a fairly high standard deviation in the marks.

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 type of part 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.

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 available (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.

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

(8 marks)

Answer pointers

A number of different fragmentation designs were acceptable involving both vertical and horizontal fragmentation based on the plant attribute. Having created the two 'plant' fragments, extra credit was given to those who showed that the presence of the plant attribute was no longer strictly necessary in either fragment.

(b) Write an SQL statement that shows how relation is fragmented in your design.

(9 marks)

Answer pointers

This was generally well answered. Candidates who knew basic SQL syntax scored heavily with three marks allocated for each of three queries that use three fragments.

(c) Justify your proposed fragmentation design for the PART relation.

(9 marks)

Answer pointers

Vertical fragmentation is performed based on the type of the plants i.e. engineering or manufacturing. As a result, examiners were expecting to see a Manchester fragment holding data (Part#, Name, Drawing#) necessary to the engineering departments while London and Hong Kong fragments would hold information (Part#, Cost, Qty) on the parts built at the local plant. Those who performed the vertical fragmentation recognised the need to include Part# in both fragments so that the original relation could be reconstructed.

Question 4

Examiners Comments

This question was attempted by around 80% of candidates and the third most popular question on the paper. The pass rate was below the overall average. Therefore the performance by candidates did not match its popularity. There are many reasons for this but it was mainly not fully answering questions was the most likely cause rather than a lack of knowledge (apart from part b).

Most of the errors were in part b) where the knowledge of SQL ALTER statements was weak. Part a) needed some precision and consistency and again some marks were lost if the matching attributes such as PerfNo had inconsistencies (ie wrong data type). Many candidates failed add NOT NULL. Part c) was poorly answered, it was insufficient to regurgitate knowledge of concurrency control but a more fuller explanation of constraints and business rules were required to gain high marks. Integrity control is managed at the application level as well as internally – where it is expected concurrency control would be handled automatically by a DBMS server.

a) Write SQL statements to **create** the Bookings, Performance and Production Tables given below. *DO NOT include foreign key references and check constraints*

Answer Pointers

CREATE TABLE BOOKINGS (TicketNo Char (6) NOT NULL, IssueDate date NOT NULL, CustNo integer NOT NULL, PerfNo integer NOT NULL, SeatNo varchar (5) NOT NULL, SeatStatus char(1), CONSTRAINT PK_Ticket PRIMARY KEY);

CREATE TABLE PERFORMANCE (PerfNo integer NOT NULL, PerfDate date NOT NULL, Production_No integer NOT NULL, Theatre varchar (25) NOT NULL CONSTRAINT PK_PerfNo PRIMARY KEY);
CREATE TABLE PRODUCTION(ProductionNo integer NOT NULL, ProductionTitle varchar (50) NOT NULL, Production Company varchar (25) NOT NULL, Sponsor varchar (25) NOT NULL);

NOTE include data types and NOT NULL

Around 5 marks were awarded for correct table definitions with 2 marks for SQL syntax and 2 marks for primary keys

b) Write ALTER TABLE statements to include foreign key references between the Performance and Production Tables.

[6 marks]

Answer Pointers

ALTER TABLE BOOKINGS ADD FOREIGN KEY (PerformanceID) REFERENCES PERFORMANCE (PerfID);
ALTER TABLE PERFORMANCE ADD FOREIGN KEY (PerfID) REFERENCES PRODUCTION (ProductionID);
[3 marks each references expression and correct PK/FK combination]

c) Explain the techniques you would use to prevent the following database operations from compromising consistency and data integrity of the application:-

- *Cancel all bookings made by a customer with name = P.Smith.*
- *Cancel all bookings for the next performance of 12th Night*
- *Change the date of a performance of a production of 12th Night from 3rd January 2011 to 4th January 2011.*[10 marks]

Answer Pointers
Cancel all bookings made by a customer P Smith – must make sure the correct member number or customer number as P Smith is not the primary key. Maybe check the address and add other info (date joined) to prevent either duplicate deletion of bookings made in this name. Business rule or check use of identity values – can corrupt the information meaning.

Need to warn the user which is the current performance and confirm whether next performance exists or is at the same theatre – the transaction potentially could delete performances in the wrong theatre. A business rule is needed.

All bookings need updating or so they get cancelled first – potential loss of integrity if transaction is not committed in one go also do customers have a choice? Could they be informed or get a refund. Is the theatre available and which theatre hosts this booking. Again define a business rule to clarify and restrict this operation on certain conditions

Question 5

Examiner's Comments

This question was attempted by around 23% of candidates and the second least popular question on the paper.

Performance was correspondingly poor compared with other questions with only 33% passing this question.

This was a concurrency question and related to how concurrency works in practice. In part a) identifying the transactions was paramount as the following parts used these transactions in the some detailed analysis of how transactions interleave and the controls built in to resolve conflicts. This is mainly text book knowledge but what was disappointing was the inability of candidates to apply this knowledge to the scenario provided.

The last part is generally not found in text books as there is trade off to be assessed and performance measures can be dependent on different set ups on DBMS servers . Full marks were given for answers that were qualified by practical knowledge of a particular DBMS server that candidates have experience.

Refer to the discourse in Appendix A.
Assume that at peak times the theatre database server processes up to 100 transactions submitted concurrently by customers accessing the theatre bookings web site.

a) List a range of CRUD (create/read/update/Delete) transactions that a particular user would submit to the theatre tables. *Hint your list should include transaction description or name , transaction type (either create/read, update/delete) , table(s) affected*

*[*5 marks]

ANSWER POINTERS
Candidates should choose a representative sample that are inferred by the generated tables populated by these transactions

Transaction	Description	Type	Tables
Check seat availability	Restrict search to date seat area and production	READ	Seats, performance, production, theatre
Confirm a booking	Search for customer provisional booking then update the	UPDATE	Seats, Booking

	status		
Cancel a booking	Change seat status to free and change booking status	UPDATE	Seats Booking
Cancel a performance	Change performance status and seat availability	UPDATE	Performance seats
Remove a seat	Remove a seat	DELETE	Seat
Create new seat plan	Generate seats for new performance and set seat status and seat codes	CREATE	Seats, Perf

b) List those (2) CRUD transactions that would conflict with transactions being processed concurrently from another user/connection. Explain how a DBMS server handles these conflicts.

[12 marks]

Answer pointers

Possible conflicts
Cancel a performance could conflict with check seat availability [1 mark]
Confirm a booking Check seat availability [1 mark]
DBMS resolves conflicts by allocating write or exclusive locks - explanation [6 marks with 2 marks for principles a and 4 marks for examples.]

c) Explain how a DBMS might be tuned to maximise high transaction throughput given the volume of concurrent transactions in the theatre booking application.

[8 marks]

Answer pointers

Use of connection pooling and caching data – physical reads is expensive so apply minimal physical reads and maximise logical reads (memory based) – in effect defer commits on disk but process as if they occurred in memory cache. Also candidates should mention relaxation of READ COMMITTED to impose a optimistic strategy this means partial reads go ahead only if X locks are not held ie defer locks to the last minute. Another technique is to impose a lower granularity of lock ie selective lock at row level – based on query optimiser execution plan to use information from a previous transaction that conveys which rows are accessed and which can be locked. marks awarded for connection pooling and caching, checking if physical reads < logical reads; changing READ COMMITTED etc and techniques mentioned in the answer pointer or equivalent in any named DBMS server.

Question 6

Examiner's Comments

This question was attempted by around 20% of candidates and was the least popular question on the paper.

Performance was correspondingly the poorest compared with other questions with only 30% passing this question.

The average mark of 7.18 also reflected the poor performance on this question.

This is central topic on the syllabus although it does not always get covered by a full question. This may mean that candidates did not expect this question and/or were unprepared. The topic is about mapping a database schema (logical level) to a higher conceptual level in this case expressed as a user interface. Clearly the conceptual level in this context is not a schema but instead a view presented for human interaction - in this case by the familiar Forms baseduser interface. To allow users to interact with logical tables the Form must present data directly from the underlying tables or represent relationships between them (such as Master detail derived from 1 to Many relationships). Most candidates succeeded in producing reasonable Forms for part a). The best answers were simple forms with clear annotation as to the purpose of each control (ie drop down) and its mapping or derived data. For part b) again very disappointing answers lacking any real insight in most cases and unable to apply a scenario to a given problem. Part c) many answers seemed very rushed and again lacked examples that applied to the scenario.

Refer to the discourse Appendix A.

a) Draft out the layout of a forms based user interface that will handle all aspects of customer booking as outlined in the discourse. Include a menu structure to navigate to the required forms. [16 marks]

ANSWER POINTERS
Any suitable interface can sketched out but it must contain field objects that allow data input for example enter perf and theatre and dates to select desired performance. Once the availability is determined, customers must confirm their membership details allowing them to book. The address details are returned from the database and the booking payment will proceed to a secure form (SSH) possibly through a 3rd party payment scheme – processing not required.
It is suggested that 6 Marks would be allocated for correctly interpreting the discourse and realising the constraints and describe the logic by walking through the steps required. Annotation of text boxes, menus are expected if there is some complex function to explain. A numbered step through of the form should also be considered.

b) Explain how your forms map to the underlying relational model and how they support the processing requirements mentioned in the discourse. State any recommendations or assumptions you made. [9 marks]

ANSWER POINTERS AND MARK SCHEME
The key concept to use are VIEWS derived tables generated by SQL code such as CREATE VIEW V1 as (SELECT … FROM … WHERE). This defines the static view of the Form and the mapped objects on the form and tabulates the mappings and data types (for validation purposes). It is also important to show how the various interactions between the form and the database is managed at code level – the question does not ask for code but expects some abstraction of the code such as names of stored procedures and parameters passed at each step eg Commit – INSERT sp whereas get perf details is a query a SP or SQL.

This part is more constrained by the discourse so judgement on marks allocation and some leeway if a) and b) has answers across these questions

c) Explain the programming techniques you would need to use when implementing your 'Forms based interface' for the theatre application. [10 marks]

ANSWER POINTERS AND MARK SCHEME

The use of Stored Procedures or Triggers running the DBMS server level should be a focus to answering this part. The client side/middle tier application have to interact using conventional

techniques e.g. ISAPI or CGI this means the code on the application server runs the client by generating HTML code dynamically as it depends on user interaction and what the database returns.

Example application code could include PHP, c# or Java with an explanation of 'code behind the forms'.

For 10 marks suggest 3 marks would cover concepts application client and data tiers and the idea of browser rendering dynamically.

Suggest around 3 marks would cover programming examples with an extra mark for examples applied to the case study

APPENDIX A: Theatre Booking Database

Discourse: The following tables represent the data tier of a client-server database application that customers access to make bookings over the WWW. The database supports the booking of seats by customers for a performance of a particular production that take place at a particular theatre. Customers can either reserve a seat or pay for a seat in full at the time of booking. Reservations must be fully paid (and hence confirmed) within 7 days of the booking if the performance is more than 14 days ahead. Following a successful booking a 'printable' ticket is generated for the customer to print off and present at the theatre when they attend the performance or if they request a refund.

Seats are organised into areas with the most expensive in the circle with individual seats identified by a seat number (eg H3) and given a seat code, such as Reserved for sponsors (code R) ; reserved for Disabled customers (code D) otherwise the seat code is NULL. Seats are also given a seat status where 'C' indicates a booking for a seat has been confirmed and paid for otherwise the seat status is NULL if the seat is reserved.

When a performances is cancelled and customers are refunded the cost of their tickets in full subject to the customer producing a printed ticket (this acts as a proof of purchase). Alternatively customers can re-book for another performance of the same production on a different date. They must re-book or request a refund within 14 days after the date of the performance.

Relational Tables

Figure A1 Bookings Table

TicketNo	IssueDate	CustNo	PerfNo	SeatNo	SeatStatus
079231	13-Dec-2010	10032	8320	H3	C
309232	13-Dec-2010	10032	8320	H4	C
309998	15-Dec-2010	3424	869321	MM3	C
306298	15-Mar-2011	3424	9767	MM3	C
736228	15-Oct-2010	3420	9770	C11	C
079232	13-Dec-2010	10035	8320	H5	
079233	13-Dec-2010	10035	8320	H6	

Figure A2: Performance Table

PerfNo	PerfDate	Production No	Theatre
8320	2-Jan-2011	1	Welldon
8321	3-Jan-2011	1	Byron
869321	15-Dec-2010	2	Byron
9767	15-Mar-2011	2	Welldon
9770	15-Oct-2010	3	Byron

Figure A3: Production Table

ProductionNo	ProductionTitle	Production Company	Sponsor
1	12th Night	RSC	Teesside Polymers
2	Chopped Carrot	Shaw-Taylor	The Vegan Society
3	12th Night	Shaw-Taylor	Gardeners World

Figure A4 Customer Table

Customer No	Customer Name	Customer Address
10032	R. Sayers	'Tess' Ilkley Moor
7243	P. Smith	**'**Homeblest', Preston Capes
10035	V. Singh	23 Belle Vue St, Odiham
3420	P.Smith	Dove Cottage Stratford

Figure A5: Seats Table

SeatNo	Seat Area	Seat Code	**Theatre**
C11	Balcony	F	Byron
MM3	Circle	D	Welldon
MM3	Stalls	D	Byron
H2	Front Stalls		Welldon
H3	Front Stalls		Welldon
H4	Front Stalls		Welldon
H5	Front Stalls		Welldon