BCS THE CHARTERED INSTITUTE FOR IT

BCS HIGHER EDUCATION QUALIFICATIONS

BCS Level 5 Diploma in IT

DATABASE SYSTEMS

September 2014

Answer **any** FOUR questions out of SIX. All questions carry equal marks

Time: TWO hours

Answer any Section A questions you attempt in Answer Book A

Answer any Section B questions you attempt in Answer Book B

The marks given in brackets are **indicative** of the weight given to each part of the question.

Only **non-programmable** calculators are allowed in this examination.

Section A

Answer Section A questions in Answer Book A

A1

Examiners' general comment: This was a very popular question. Almost all candidates attempted it with a pass rate of 67%.

(a) A company stores employee records in the format shown below. Here we show two such employees.

Employee ID: E05	**Name:** Fred Shred	**Job Title:** Manager
Branch Code: B04	**Branch Name:** London	
Qualification	**Level**	**Year Obtained**
BSc	Undergraduate	1986
PhD	Postgraduate	1990

Employee ID: E07	**Name:** Jim Brown	**Job Title:** Assistant
Branch Code: B04	**Branch Name:** London	
Qualification	**Level**	**Year Obtained**
BSc	Undergraduate	1995

(i) Identify the repeating group of attributes and transform the above unnormalised table into tables that are in 1st Normal Form.

(5 marks)

(ii) Identify any partial dependencies and transform into tables that are in 2nd Normal Form.

(5 marks)

(iii) Identify any transitive dependencies and transform into tables that are in 3rd Normal Form.

(5 marks)

Answer Pointer

(i)
(Qualification, Level, Year) is a repeating group. (2 marks)
The following tables are in 1NF:
Employee(EmployeeID, Name, JobTitle, BranchCode, BranchName) (1 mark)
Qualified(EmployeeID*, Qualification, Level, Year) (2 marks)

(ii)
partial dependency:
Qualification → Level (2 marks)

Employee(EmployeeID, Name, JobTitle, BranchCode, BranchName)
Qualified(EmployeeID*, Qualification*, Year) (1 mark)
QualLevel(Qualification, Level) (2 marks)

(iii)
Transitive dependency:
EmployeeID→BranchCode→BranchName (2 marks)

Employee(EmployeeID, Name, JobTitle, BranchCode*) (1 mark)
Branch(BranchCode, BranchName) (2 marks)
Qualified(EmployeeID*, Qualification*, Year)
QualLevel(Qualification, Level)

Examiner's Comment: Many candidates failed to identify the repeating group of attributes and, hence, carried forward this mistake to the next parts of the question. Some confused partial dependencies with transitive dependencies. Others seem to be able to write the final set of relations straight from un-normalised form into 3rd normal form but without going through each step of normalisation.

(b) A company lists staff rooms and telephone extensions in the table below. Each room has a single telephone extension.

Staff

StaffID	**Name**	**Room**	**Extension**
S001	Smith	A7	3142
S002	Jones	B3	4500
S003	Brown	A7	3142
S004	Khan	B4	4501

(i) Explain what is meant by an *anomaly* in a table.

(2 marks)

(ii) List three kinds of anomaly, and give an example of each with reference to the above table.

(6 marks)

(iii) Explain what needs to be done in order to remove any potential anomaly in the table.

(2 marks)

Answer Pointer

(i) An anomaly is a situation where inconsistent data is introduced to a table or data is unintentionally lost. (2 marks)

(ii) (2 marks each)

- **Update anomaly: e.g. Extension for A7 changes, but is only updated for Smith.**
- **Insert anomaly: e.g. 'Singh' added to room B3, but extension entered incorrectly. (Or inability to add room that has not yet been occupied.)**
- **Delete anomaly: e.g. S004 is deleted, losing only record of room B4, extension 4501.**

(iii) The table needs to be split in order to remove the transitive dependency:

Staff(StaffID, Name, Room*) (1 mark)

Room(Room, Extension) (1 mark)

Examiner's Comment: The definition of anomaly should capture the type of problem (inconsistency and loss of data) generated by the three types of anomalies. Most candidates were able to provide an example of an update anomaly, but not so much for insert and delete anomalies. For example, a typical mistake is to assume that a new staff cannot be inserted if they do not have a room or extension. A record can be inserted as long as there is a value for its primary key (staffID). Moreover, many candidates did not provide an adequate answer to part (iii) of the question, simply talking in general terms about the fact that tables need to be normalised. The answer should be specific to the context of the table provided in the question.

A2

Examiners' general comment: A fairly popular question attempted by nearly half of all candidates. Average mark was a little disappointing but in line with expectations. There was a small improvement in performance of candidates when compared to previous questions of this type in previous years.

Refer to Figures A1.1 and A1.2 listed below representing a Tool Hire database.

a) Identify and explain the SQL code that implements 'referential integrity'**.**

Answer Pointer

The Hire table has a common field (CustID) that relates to the Hirer so that the Hirer for a Hired tool must exist prior to creating a instance/row in the table Hire
The section of the script that identifies referential integrity occurs once and is listed below

```
CONSTRAINT FK_hire_customer FOREIGN KEY(CustID)
  REFERENCES customer (CustID),...);
```

(4 marks)

Examine'sr comment: Generally well answered no real problems.

b) What order must these Tables be populated in order to preserve referential integrity?

Answer Pointer

Hire Rate , Tool, Tool Accessory, Hirer Hire

(3 marks)

Examiner's Comment: Poorly answered in general. The Hire table for example must come last as it has dependencies on Hirer,Tool for example. If the correct order is not obeyed then Referential Integrity could break down.

c) Identify and explain the effect of three further data integrity constraints present in the script. {Hint think about how these constraints are invoked)

Answer Pointer

Primary key entity integrity, Check and NOT NULL.
Insert statements are checked for uniqueness in PK, correct value range (check constraint) and automatically rejection of inserts that attempts to place a null in column defined as NOT NULL .

(6 marks)

Examiner's Comment: Generally well answered as most candidates identified 2 of the 3 – the majority missed the NOT NULL constraint. Others that were mentioned (and gained credit) included data type checks and 'char' data type widths represented in some answers.

d) Using the tables in fig A1.1 explain the function of each of the following SQL keywords using sample SQL queries (or a single SQL query if you prefer). Show what output is produced by running each of your queries (query).

```
COUNT
GROUP BY
HAVING
INNER JOIN
IN
LIKE
```

Answer Pointer

A single SQL query could be given as an answer though although some flexibility is allowed.

```
SELECT COUNT(quantity)
    ,CUSTOMER.CustID
FROM HIRE INNER JOIN Customer
ON CUSTOMER.CustID = HIRE.CustID
WHERE CustSname LIKE 'B%'
AND CustTitle IN ('Mr','Dr')
GROUP BY CUSTOMER.CustID
HAVING COUNT(*) >1;
```

Output for the above query would be

```
COUNT(QUANTITY)  CUSTID
---------------- ------
              2   2127
```

(12 marks)

Examiner's Comment: Generally well answered as most candidates supplied individual examples of SQL to describe each keyword. There are many candidates though that fail to realise that the GROUP BY clause should contain the non-aggregated columns listed in the select list.

Fig A1.1 Tool Hire database sample data

(note Primary Keys are in bold and Foreign Keys are in italics).

TOOLAccessory

ToolID	AccessoryName
3215	Blades
3225	7a PowerPack
3299	Bit set
3377	Sanding Disc
3377	Polishing Disc
3377	7a PowerPack

TOOL

ToolID	ToolName	*ToolCat*
3215	Circular saw	Joinery
3299	Drill	Building
3371	Hammer	Misc
3377	Sander	Decorating
3379	Wallpaper Stripper	Decorating

HIRERATE

ToolCat	HireRate	Deposit
Joinery	31	50
Building	31	60
Decorating	20	40

CUSTOMER

CustID	CustSname	CustTitle	CustPhone
43	Mogul	Mr	07422971
51	Dannisake	Ms	06454853
127	Borge	Dr	16468191

HIRE

ToolID	***CustID***	HireStart	HireEnd	Quantity
3225	43	13/02/13	14/02/13	1
3377	43	13/02/13	15/02/13	2
3377	51	14/02/13	15/02/13	1
3299	127	26/02/13	29/02/13	2
3299	127	28/02/13	29/02/13	3
3229	51	28/02/13	29/02/13	1

Fig A1.2 SQL script used to (partially) create Tool Hire tables.

```
CREATE TABLE tool (ToolID INT NOT NULL PRIMARY KEY,...);

CREATE TABLE hire(ToolID INT NOT NULL,
                  CustID INT NOT NULL,...
                  Quantity NUMBER DEFAULT(1),
       CONSTRAINT PK_hire PRIMARY KEY ToolID),
       CONSTRAINT FK_hire_customer FOREIGN KEY(CustID)
       REFERENCES customer (CustID),...);

CREATE TABLE toolaccessory(ToolID int NOT NULL,
                           AccessoryName VARCHAR(25) NOT NULL,
       CONSTRAINT PK_Toolsaccessory PRIMARY KEY
                        (ToolID,AccessoryName),...);

CREATE TABLE customer(CustID INT PRIMARY KEY,...);
```

```
CREATE TABLE hirerate(ToolCat VARCHAR(25)...
                      Deposit NUMBER(5,2)
                              CHECK (deposit > 0 AND deposit < 50));
```

A3

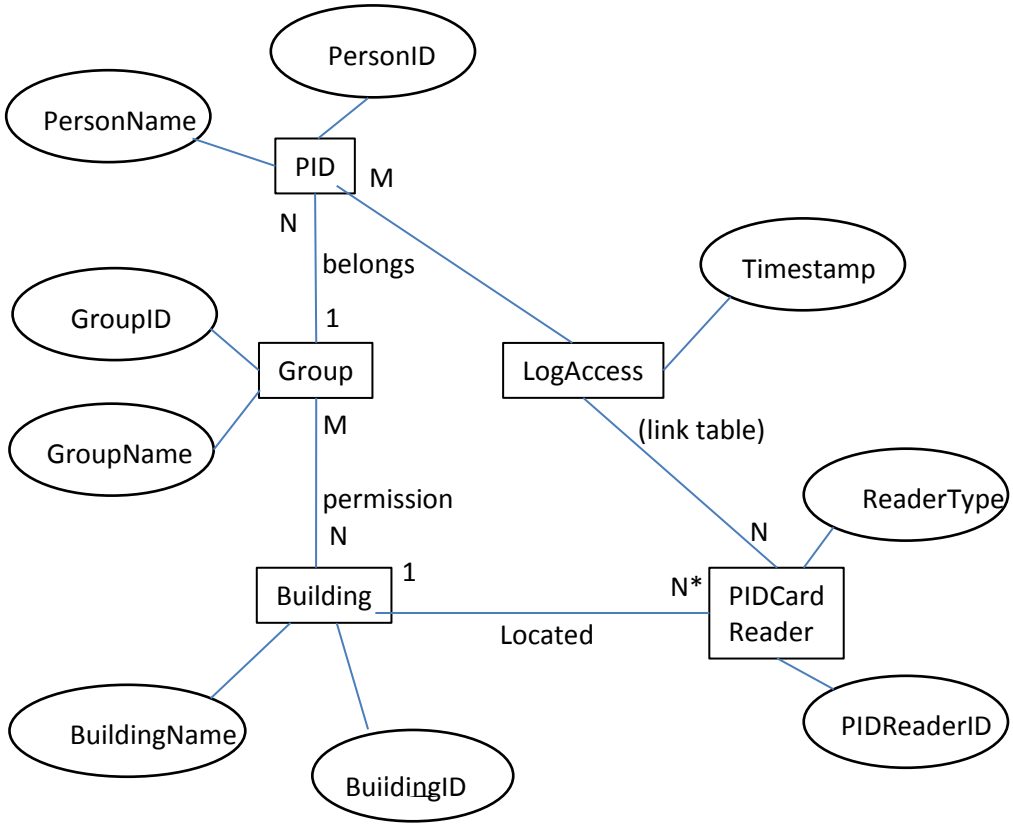
Examiners' general comment: The third most popular question having been attempted by 73% of candidates. Overall a mixed bag of results but in general most candidates came quite near to a representative ER model close to the desired solution.

A University has adopted a personal identity card (**PID**) system to improve security and to restrict access to certain **groups** of people (such as students, teachers, professors , secretaries, managers etc) and at certain times and dates. A person is issued a PID card as soon as they become part of the University community (either employed or on a course of study). Each person belongs to only one group which determines what buildings they can access.

To enter a building, a person (each having a unique **PID**) must have permission which is established when their PID card is swiped through a PID card reader outside the **building** they wish to enter. **PID card readers** are located outside the door of a building users wish to access. Permission is granted only if their access credentials are successful. If access is allowed, the captured data is logged, recording the date, personID (from the PIDcard) and the PIDreaderID.

a) Produce an ERD of the above scenario using the highlighted keywords as Entity Types. State the notation you used and include in your ERD model :
   - Cardinality expressed as either 1:many or many to many.
   - Primary keys and sample attributes.
   - Any assumptions that do not contradict the discourse

Answer Pointer
Minimal notation cardinality given below



Cardinality/asumptions

- **An ID card belongs to one individual who belongs to one distinct (eg staff,student etc) group.**
- **A Group is associated with many PIDcards**
- **A PIDreader is located in one building and each building may contain one or many PID readers**
- **A PIDCard is read by one PID reader on a particular date and a PID reader validates many PID cards**
- **Permission to access is determined by referencing group membership to building.**
- **Each unique logaccess monitors both success and denial based on an individual persons membership of a group. Hence :-**
    - **Entity PIDCardReader – it would be useful to add an attribute say readertype or status.**
    - **The Relationship between Building and PIDcardReader has a cardinality of 1:N -this assumption needs stating by candidates as an alternative assumption could be that this relationship's cardinality is 1:1. Discourse does not state either way. It is realistic to assume that either interpretation could be made.**
- **An attribute that monitors logAccess indicates 2 states - either denial or success of access is recommended**

(15 marks)

Examiner Comment:
Generally fairly well answered as many candidates obtained the correct relationships. Most candidates gained around 10 marks for this question if they missed one or two relationships and attribute definitions. Overall there were few variations to the base ERD solution above and some of these were valid interpretations especially if they were backed up with assumptions.

b) Derive from your ERD a set of normalised tables populated with sample data

(10 marks)

ANSWER pointer

PIDCard

PID	GrpID	PName
787	4	FredBloggs
788	4	JonSmith
789	1	MaryBrown
799	5	GeorgeTan
800	5	JohnSmith
801	5	AnnMann
802	3	JaneHutt
880	2	TobyHall

Building

BID	BuildingName
1	Europa
2	Enterprise
3	Victoria

Group

GrpID	GrpName
1	Secretary
2	Admin
3	Professor
4	Teacher
5	Student

Permission

PermID	GroupID	BID
1000	4	1
1001	5	1
1002	5	1
1003	3	1
1004	3	2
1005	1	2
1006	2	3

PIDreader

PIDReaderID	ReaderType	BID
231	standard	1
232	smart	1
233	standard	2
234	standard	3

LogAccess

AccessID	PIDReaderID	PID	TimeStamp
987	232	787	11/11/13:11:22:54
988	232	799	11/11/13:11:23:55
989	233	789	11/11/13:12:24:30
990	234	880	11/11/13:12:59:55

Examiner's Comment: Generally not as well answered as expected given the overall success of producing an ER model. Many candidates seem to struggle in mapping an ERD to logical tables and hence the supplied sample tables did not achieve consistency with the ERD. Many candidates lost marks by failing to populate the sample tables and those candidates that went through normalisation to derive the tables undertook unnecessary work with little reward. A correctly formulated ERD will normally lead to the construction of tables in 3NF.

Database Systems (Part B)
B4

Examiners' general comment: A reasonably popular question having been attempted by 54% of candidates. Around 47% of attempts achieved pass level marks

This question uses the 'Professionals' relation below…

Professionals

ID	Name	Profession	Age	Salary
1	Billy Builder	Architect	45	75,000
2	Steve Surfer	Swimming Instructor	21	13,000
3	Frankie Fetch	Lorry Driver	33	24,000
4	Brian Brush	Dentist	45	75,000
5	Nicky Nurse	Midwife	38	32,000
6	Roger Road	Lorry Driver	27	25,000
7	Fiona Floss	Dentist	52	120,000
8	Colin Crawl	Swimming Instructor	24	13,000

(a) Based *solely on the 'Professionals' relation*, write down the answer to the following calculation…

$$\frac{\text{(DEGREE x CARDINALITY) + (NUMBER OF DOMAINS)}}{\text{(NUMBER OF CANDIDATE KEYS)}}$$

Hint: For full marks you must show all working out and all intermediate figures.

Answer pointer

DEGREE = 5, CARDINALITY = 8, DOMAINS = 5, CANDIDATE KEYS = 2

Final answer = ((5 x 8) + 5)/2 = 22.5

(5 Marks)

Examiner's Comment: Surprisingly poorly attempted. While most candidates who attempted this sub-question managed to get various aspects correct (such as the degree or number of domains) only a handful got all four concepts correct and hence the right final answer.

(b) *Using the 'Professionals' relation and Venn diagrams (or any other suitable diagrams)*, explain how the following relational algebra operations are processed. You must explain the *key concepts*, provide an *actual example* and a *suitable diagram* for each.

(i) UNION

(3 Marks)

(ii) INTERSECT

(3 Marks)

(iii) MINUS (DIFFERENCE)

(3 Marks)

(iv) SELECTION

(3 Marks)

(v) PROJECTION

(3 Marks)

Answer Pointer

To get the full three marks, each RA operator must have the key *concept* explained (for example, UNION is the merger of all sub-sets with duplicates removed), a *diagram* (Venn or otherwise) and a simple illustrative *example*.

Examiner's Comment: General descriptions, examples and Venn diagrams of the first three operations were of a good standard, but candidates struggled with the final two. Most candidates scored well in this section, falling down only on the SELECTION and PROJECTION parts.

(c) For each of the following relational concepts, explain the key ideas behind it, provide a suitable example and/or diagram and compare/contrast their application.

- UNION COMPATIBILITY and its importance for set operations
- JOIN CRITERIA and its importance for seeing if two or more tables are joinable

Answer Pointer

UNION COMPATIBILITY

Each set to be operated upon must be comparable and have like-for-like compatibility. This means that each relation or results set to be operated upon must have the same number of attributes and comparable attributes are defined over a similar domain.

JOIN COMPATIBILITY

With the exception of natural joins, the actual names of the attributes are irrelevant. It is the domains that matter. For two or more relations or tables to be joinable they must share at least one common attribute, whereupon they may be merged, row-by-row, to form a merged (joined) result – which may be further processed or not.

Good diagrams would earn bonus marks.

(5 Marks)

Examiner's Comment: Generally very poorly attempted – with many vague and imprecise descriptions of both concepts. The number of students confusing UNION COMPATIBILITY with UNION was much less than in previous years but still, the preciseness of response was weak.

B5

This question was the least popular question on the paper and was attempted by 41% of the candidates. The spread of marks was quite wide with some extremely good attempts.

(a) Using your own examples, describe the various different types of *user interfaces* and *development environments* that may be employed to access a database, taking care to discuss the key features, strengths and limitations of each.

Answer Pointer

Points similar to the following would gain credit…

- **Text-based (command-line) versus GUI based**
- **Form-based versus web-based**
- **Technical users versus end users**
- **Specific examples of each – APEX, SQL*Plus, SQL Developer, TOAD, OEM etc**

(10 Marks)

Examiner's Comment: This invoked a wide range of responses. Many students picked up on the fundamental differences between command-line, GUI and web-based interfaces, making good comparisons of their relative strengths & weaknesses (ease of use, the need for syntax knowledge etc). A smaller percentage drilled down to talk about form and report generators, IDE platforms for developers and specialized DBA tools such as Oracle's Enterprise Manager. A few students raised the issue of touch-screen interfaces – which while not expected was rewarded. Overall, good general distinctions raised but most responses lacked depth.

(b) Describe, with the aid of a well annotated diagram, how and where the following three concepts are implemented across a three-tier web-database architecture. You should take care to clearly explain the key concepts, potential problems and implementation options available for each.

(i) Persistent, stored data

(5 Marks)

(ii) Presentation of output

(5 Marks)

(iii) Business logic

(5 Marks)

Answer Pointer
Points similar to the following would gain credit…

- **Data – held under the control of the 'back end' DBMS. Main issues are data integrity and data security plus query optimization for larger data sets. The role of the DBA. Examples could be Oracle, MySQL etc.**
- **Presentation – the end-user view – typically via a web browser or other 'front end' interface. Main issues are usability and data validation to avoid invalid data reaching the back-end database and so wasting network traffic – via use of drop-down lists, on screen help, calendars, radio buttons, client-side scripting such as JavaScript etc.**
- **Logic – business logic/application logic held in the middle tier – the glue that connects the front-end user to the back-end data. Various web technologies and server-side scripting languages are available - such as PHP - to formulate database queries and to render the resulting web pages for the browser.**

Marked holistically but a clear, well annotated diagram is essential for higher marks.

Examiner's Comment: Generally poorly and vaguely answered with poor-quality (or missing) diagrams. Many candidates conflated the ANSI-SPARC 3-level architecture of a DBMS with the wider-scale 3-tier architecture of a web-database infrastructure. Most candidates did well when explaining the back-end database tier (DBMS) and to a lesser degree the front-end presentation layer but the middle layer concerned with business/application logic was very poorly explained.

B6

Examiners' general comment: 77% of candidates attempted this question with a pass rate of 53%.

a) Transaction processing is one of the main features of any DBMS. Explain the following concepts:
   - Transaction
   - Transaction scheduling
   - Serializability

Answer pointer

Transaction: A transaction is concerned with accessing or changing the contents of a database. It has a clear start and finish. The assumption is that it will start its execution on a consistent database and if it runs to completion then the transaction is said to be committed and the database will be in a new consistent state. Transaction has ACID properties.
Transaction scheduling: Scheduling of transactions is under the control of a transaction manager and may involve different users submitting transactions that execute concurrently; accessing and updating the same database records. The transaction manager is responsible for interleaving different transactions. If transactions use the same database items, there are various ways in which a transaction, though correct in

itself, can nevertheless produce the wrong answer or can leave a database in an inconsistent state because of interference on the part of some other transaction.

Serializability: If a set of transactions executes concurrently, we say that the schedule is correct if it produces the same results as some serial execution. The objective of serializability is to find nonserial schedules that allow transactions to execute concurrently without interfering with one another. The result should be the same as if the database state had been produced by a serial execution. Serializability identifies those executions of transactions guaranteed to ensure consistency. It is essential to guarantee serializability in order to ensure database correctness.

Examiner's Comment: Most candidates managed to give an appropriate description of a transaction and transaction scheduling, however many struggled with the idea of serializability.

(8 marks)

b) Describe two kinds of problems (giving examples) that might appear when transactions run concurrently.

Answer Pointer

Explanation plus example for two problems such as: (4 marks x 2)

- **Lost update problem - Two transactions access the same database items and have their operations interleaved in a way that makes the value of some database item incorrect. – plus example**
- **Uncommitted dependency problem - Occurs when one transaction can see intermediate results of another transaction before it has committed. – plus example**
- **Inconsistent analysis problem. - Occurs when a transaction reads partial results of other incomplete transactions – plus example**

Examiner's Comment: Most candidates are able to list two types of problems that can arise when transactions run concurrently. However, some of the examples used show that there is a lack of understanding of these problems.

(8 marks)

c) A university is considering using spreadsheets to record the details of its students and courses. Discuss the disadvantages of this solution and whether it would be best using a database.

Answer Pointer

Two marks for listing one disadvantage and discuss it in the context of the university scenario. The answer should include at least four issues such as: Concurrency control, integrity management support, security, recovery services

Examiner's Comment: Most candidates were able to list some of the common disadvantages of using a file-based system but not many were able to discuss these disadvantages in this particular scenario.

(7 marks)