BCS THE CHARTERED INSTITUTE FOR IT

BCS HIGHER EDUCATION QUALIFICATIONS
BCS Level 5 Diploma in IT

DATABASE SYSTEMS

Tuesday 31st March 2015 – Morning

Answer **any** FOUR questions out of SIX. All questions carry equal marks
Time: TWO hours

Answer any <u>Section A</u> questions you attempt in <u>Answer Book A</u>
Answer any <u>Section B</u> questions you attempt in <u>Answer Book B</u>

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

Calculators are **NOT** allowed in this examination.

Section A

**Answer Section A questions in Answer Book A**

A1

EXAMINER'S OVERALL COMMENTS
This was a popular question with two thirds of the candidates making an attempt.
Despite this, the average mark was slightly less than expected given the prominence of
this type of question in previous papers. The reasons why candidates score lower marks
than expected, is probably down to lack of practice in data modelling. Candidates must
be more engaged in data modelling through work and tutoring.

Refer to the ER model given in Figure A1. Assume that :-

A Trainer trains one or many horses and a horse is trained by no more than one Trainer.
A given horse competes in a particular race ridden by a particular jockey on a specific date and
the position (ie 1st,2nd etc) in which a horse finishes is recorded.

Figure A1 ER model (UML Class diagram notation) for use in question A1
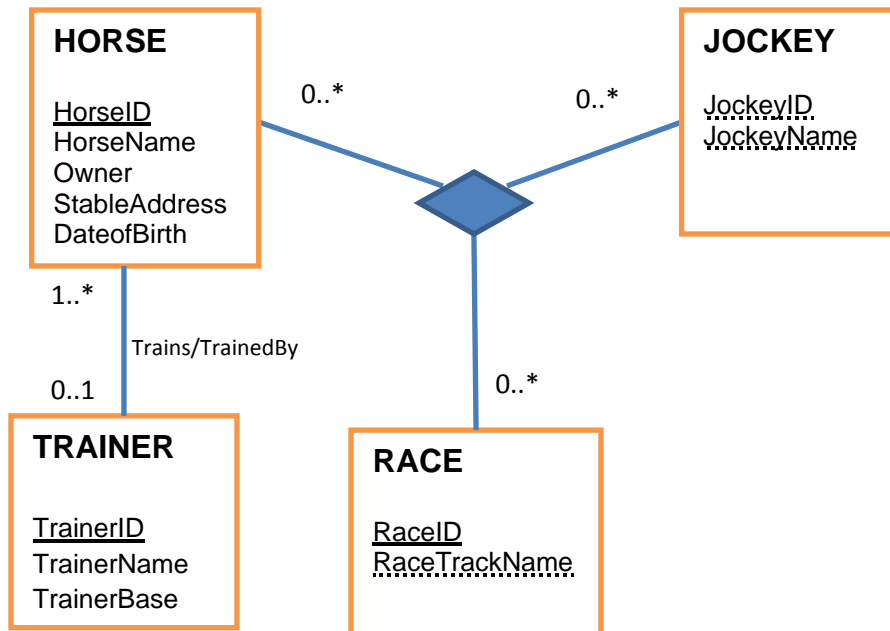
a)   In the context of database design give a precise definition of an Entity Type.   **(2 marks)**

ANSWER POINTER
Entity type describes an object, a noun, a class any meaningful collection of objects of the same type having the same properties/attributes

EXAMINER'S COMMENTS
Most candidates had a good overall knowledge of this term showing it is well known and distinctive.

b) Outline why an ER model assists in database design.   **(5 marks)**

ANSWER POINTER
High level model useful for semantic extraction and conveyance of meaning about real world objects and how they are associated. ER model often assists in normalisation if the number of tables is large as it is relatively easy to transform an ER model to 2NF. Many Case tools translate ER models to relations so an ER model can be regarded as a blueprint for many automation tools in DB design.

EXAMINER'S COMMENTS

Again most of the above points were covered with a deviation in answers to the actual question that emphasised benefits rather than techniques that assist designers of ER models.

c)   Explain using examples from the ER model (figure A1) the difference between a *ternary*   relationship and a *binary* relationship.   **(5 marks)**

ANSWER POINTER
One mark each for identifying these relationships. Ternary relationships have three participating entity types whereas binary have two. Example : results and trains respectively. An example will show diagrammatically examples of instances of horseID,jockeyID,RaceID all participating in the same relationship. Binary relationship instances have a set of instances that relate a horse to a particular trainer and a trainer related to many horses. Some instances of a horse may not have a (current) trainer.
The accuracy of the cardinality and participation is important in both cases.

EXAMINER'S COMMENTS
Most candidates could successfully identify the different types of relationship although a few confused the term with the cardinality of a relationship. Rather than redraw the ERD the answer required some explanation of what these types of relationship looked like at the instance level. Very few candidates actually used diagrams as a succinct way of achieving this, instead much longer written explanations were attempted.

d)   S
tate why many to many relationships need to be resolved into one to many relationships in an ER model.   **(2 marks)**

ANSWER POINTER
The relational model derived from an ER model cannot accommodate many to many relationships only one to many and one to one relationships. This is because a many to many relationship hides attributes that would suggest the existence of an entity type instead of the relationship type. Further the new entity type might represent a notational table (linking table) that when mapped to Tables consist of posted identifiers from the participating entity types.

EXAMINER'S COMMENTS
**Very few candidates could succinctly express the main reason, instead may candidates resorted to waffle by describing some superfluous or secondary**

reasons in contrast to what was required. Many candidates also did not realise this part is connected to preceding and following parts that should direct them to an acceptable answer.

e)   E
xplain how you would modify the ER model given in Figure A1 in order to resolve many to many relationships.   **(6 marks)**

ANSWER POINTER

Figure A1.1 ER model (UML Class diagram notation) for use in question A1
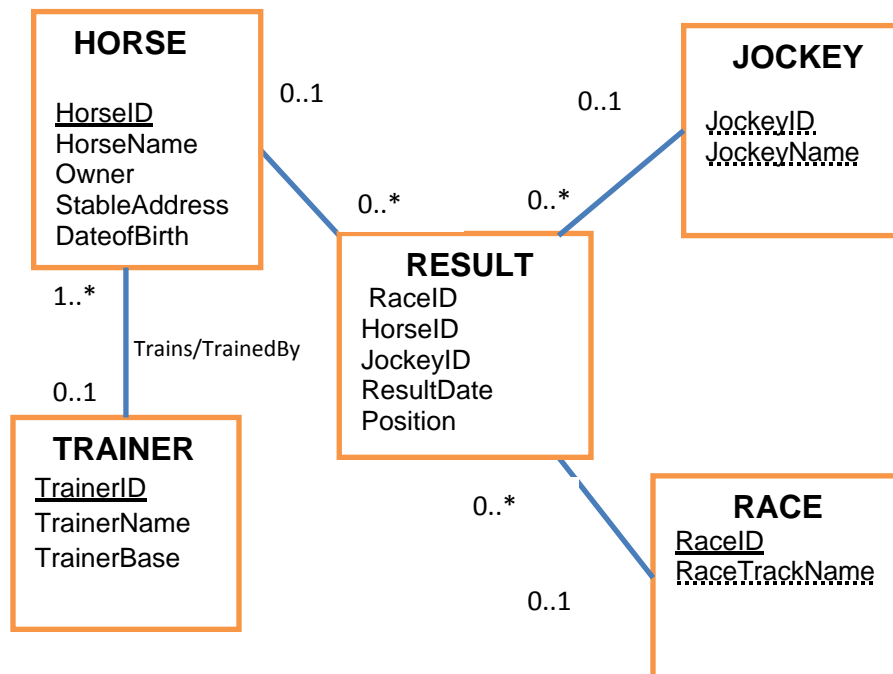
Note: The model requires the introduction of a new entity type Result replacing the relationship type of the same name. Result date is part of the identifier as the same race might occur more than once. The cardinality changes as shown to reflect the new 1..* relationships between the new entity type Result Further attributes such as place (finishing position), betting, may be introduced. Assume a race is undertaken at a the same track on different dates hence the introduction of result_date.

EXAMINER'S COMMENTS

This part was answered fairly well, but only about a half of candidates appreciated the need to introduce only one new entity (result). Some candidates did not realise this (easy) solution and as a consequence reorganised the ternary relationship into 3 binary relationships negating the succinctness in the model already

provided and introducing fantraps. The one to many relationship between Trainer and horse was already resolved but some candidates failed to recognise this and again made more unnecessary work for themselves by interpreting this as a many to many relationship. The accuracy of the resultant one to many relationships was important and some candidates failed to realise this in their answers and lost marks as a consequence.

f)   Explain how you would translate your modified ER model (in part d) as a set of Tables, giving table names, column names and an indication of primary and foreign keys.

(5 marks)

ANSWER POINTER

Many to many relationships automatically introduce an associative entity that maps directly to a new table with a composite key consisting of fields that exist in the original tables forming the relationship. Hence the ternary relationship would be transformed into a table with a composite key comprising of the three identifiers from the three entity types forming the relationship. One to Many relationships map to two tables generally although technically this usually depends on the participation constraint on the one side.

Result Table   (note italics/underlined = Foreign Key /bold = Primary Key.
HorseID.JockeyID, ResultDate, **RaceID**, Position

Horse table
HorseID, horsename ,Owner, DateOfBirth, StableAddress, *TrainerID*

Trainer table
TrainerID , TrainerName, TrainerBase...

Jockey table
jockeyID, jockeyName

EXAMINER'S COMMENTS
This part was dependent on the previous part and hence some marks were lost because of the uncertainty in dealing with the ternary relationship and its mapping. Also it was concerning to see in many cases the absence or incorrect placement of the foreign key *trainerID* in the *horse* table without a valid explanation. Clearly every horse has one trainer and a trainer can train one or more horses not the other way round as many candidates assumed. It might be candidates were unfamiliar with the UML notation and certainly some were unfamiliar with the mapping rules.

A2

EXAMINER'S OVERALL COMMENTS
Not a very popular popular question with just under a half of candidates attempting it. SQL is a key language and a popular and practical topic in this exam. Despite this many candidates reveal a lack of practical experience which is key to being successful in this topic. All the questions challenged candidates to reveal knowledge of either understanding or applying frequently used SQL keywords and clauses to supplied data. Particular knowledge of stored procedures and use of HAVING in GROUP BY was weak.

Refer to the following tables then answer the questions that follow.

Table1 Publisher

```
pub_ID pub_name          address         city        state
------ ----------------- --------------- ---------- -----
736    New Age Books      4 1st Ave.      Boston      MA
877    Binnet&Barney     23 34rd St.      Washington DC
1389   AlgoDumini        47 9th Ave.      Berkeley    CA
```

Table2 Titles

```
title_ID title             cat        pub_ID price
-------- ----------------- ---------- ------ -----
2091     is anger the enemy? psychology 0736   10.00
2106     life without fear  psychology 0736    7.00
1035     alien              sci-fi     1389   19.99
```

a) Show the data output produced when each of the following SQL queries are executed.

SQL query 1:

```
SELECT pub_ID,pub_name
FROM publisher
WHERE state NOT IN ('MA','IL')
AND pub_name NOT LIKE 'A%'
```

SQL query 2:

```
SELECT cat ,
AVG(price) AS AveragePrice,
SUM(price + price*0.2)  AS TotalPrice
FROM titles
GROUP BY cat;
```

(8 marks)

ANSWER POINTER

query 1

```
PUB_ID PUB_NAME
------ --------------------
877    Binnet and Barney
```

query 2

```
CAT         AVERAGEPRICE      TOTALPRICE
----------  ----------------  --------------------------
psychology 8.500              20.400
sci-fi     19.990             23.988
```

EXAMINER'S COMMENTS

Candidates with a reasonable knowledge of SQL (the standard database query language) should have scored heavily, but unfortunately this was not always the case. The problem with aggregate operations and understanding GROUP BY seems to continue to confuse. Also an accurate answer was required and thus candidates must remember to include the headers of the result set. Slight errors in the calculations of averages and sums were permitted.

b) What is the purpose of the HAVING keyword? Use a HAVING clause in query 2 above to show how it can be used to limit the rows in the output :-   **(4 marks)**

ANSWER POINTER

HAVING qualifies a GROUP BY with a condition operating on the group/set defined such as

```
SELECT cat
     ,AVG(price) AS AveragePrice
     ,SUM(price + price*0.2)  AS TotalPrice
FROM titles
GROUP BY cat
HAVING MIN(price) > 10.99
```

for example would return

```
CAT         TOTALPRICE    SETL
----------  ------------- --------------------
sci-fi     23.988         19.990000
```

EXAMINER'S COMMENTS

This part was poorly attempted with only about a fifth of candidates realising the HAVING keyword imposes a constraint on a resultant computation as indicated above. Most candidates did not appreciate that a condition imposed on an aggregate operation must be incorporated in a HAVING clause.

c) Discuss the advantages of using stored procedures as a way of implementing database queries. **(7 marks)**

ANSWER POINTER

Stored procedures encapsulate frequently run segments of code written in SQL code. They support parameterisation and may include sections of SELECTS/INSERTS/UPDATE statements.

The code can be the same but the results are different by using different parameters and conditional expressions (IF/THEN/ELSE etc) when the code is run. Key advantages are reuse of code; security; abstraction/hiding code; avoid SQL injection attacks; validation code close to database.

EXAMINER'S COMMENTS

The absence of informative answers or any answers at all indicates a poor knowledge of stored procedures. There are of course many variations in the way stored procedures are supported in a DBMS. But needless to say they are widely supported by all major DBMS vendors and candidates should be more familiar with them during their work/learning activities.

d) Write SQL code to convert SQL query 1 into a View and explain the advantages of using Views as a way of implementing database queries. **(6 marks)**

ANSWER POINTER

```
CREATE VIEW currentv (
      Types
     ,val1
     ,val2 ) AS
SELECT cat,
SUM(price+ (price*0.2)) ,
AVG(price) AS setl
FROM titles
GROUP BY cat;
```

The advantages of Views are mainly for security and convenience and the ability to use a virtual table to break down complex queries into stages.

EXAMINER'S COMMENTS

Most candidates were familiar with views and answers reflected this. Many candidates missed the AS keyword in the view definition.

A3

EXAMINER'S OVERALL COMMENTS

This was an extremely popular question with almost all candidates attempting it and achieving relatively high marks.

a) The table shown below stores details of students and the overall grade each student obtained in different modules. The Primary Key is (StudentID, ModuleID).

Results

StudentID	StudentName	ModuleID	ModuleName	Grade
S001	Smith	M01	Java	A
S001	Smith	M02	Databases	B
S002	Ford	M01	Java	B

(i) Which Normal Form does the above table violate and why?   **(3 marks)**

(ii) Give an example of an update anomaly and an example of a delete anomaly that may occur if the table is left un-normalised. Explain the problems that are caused.

(4 marks)

(iii) Show how you would normalise the table.   **(5 marks)**

ANSWER POINTER

(i) The table violates 2nd Normal Form (1 mark) because there are two partial dependencies: StudentID $\rightarrow$ StudentName and ModuleID $\rightarrow$ ModuleName (2 marks)

(ii) Update anomaly: if a module is to be renamed, it is possible to mistype one of its instances, thus introducing an inconsistency (2 marks).

Deletion anomaly: if students are deleted, modules will also be deleted, resulting in a loss of modules' details (2 marks).

(iii)

Student(StudentID, StudentName)   **(1 mark)**

Module(ModuleID, ModuleName)   **(1 mark)**

Results(StudentID*, ModuleID*, Grade)   **(3 marks)**

EXAMINER'S COMMENT

Most candidates managed to correctly normalise the given table in step (iii), however, not all were able to provide a good example of update/delete anomalies. Also, some candidates confused partial dependencies with transitive dependencies.

b)   An important concept in the theory of relational databases is that of a *functional dependency*.

(i)   Explain what is meant by a functional dependency and give an example.   **(2 marks)**

(ii) Identify two functional dependencies in the following table (A, B and C are the attributes):

A	B	C
a1	b1	c1
a1	b1	c3
a1	b2	c1

(4 marks)

ANSWER POINTER

(i)   An attribute B is functionally dependent on an attribute A if each value of A is associated with a value of B (1 mark). For example, StudentID $\rightarrow$ StudentName (1 mark).

(ii)   B $\rightarrow$ A (2 marks) and C $\rightarrow$ A (2 marks)

EXAMINER'S COMMENT

This was a question with which almost all candidates struggled. Very few managed to correctly identify the two dependencies in (ii), and no one managed to provide a sound explanation of functional dependency.

c)   A company uses the table below to record details of staff. Each staff has up to three qualifications:

StaffID	StaffName	Qualifications
S01	Ibanga	BSc, MSc, PhD
S02	Kumar	BSc, MSc
S03	Grant	BSc, PhD

(i) Explain why this table is not in "First Normal Form" (1NF). **(1 mark)**

(ii) Show how this table can be transformed into 1NF tables.

Give **two** possible solutions. **(6 marks)**

ANSWER POINTER

(i) The attribute "qualifications" is multi-valued on the PK, hence the table is not in 1NF. (1 mark)

(ii)
Solution 1:
Students (StaffID, StaffName, Qualification1, Qualification2, Qualification3)(2 marks)

Solution2:
Students (StaffID, StaffName)   (2 marks)
Qualifications (StaffID*, Qualification)   (2 marks)
Alternatively, students can use a third table (QualID, Qualification)

EXAMINER'S COMMENT

Almost all candidates recognised the need to split the table as a way of normalising it. However, very few were able to think of a different solution where new columns could be added to store the qualifications.

Section B

Answer Section B questions in Answer Book B

B4

EXAMINER'S GENERAL COMMENTS
A very popular question with three quarters of candidates attempting it – many achieving high marks.

(a)   With reference to a sample relation of your own choosing, explain and discuss the following relational model terminology, including its function and any related concepts.
A good diagram showing your sample relation is strongly suggested.
- Tuple
- Attribute
- Domain
- Degree
- Cardinality
Each item is equally weighted.   **(15 marks)**

ANSWER POINTER
Three marks per item with full marks only for covering all the following points. Tuple means row. Each row is uniquely identified by a primary key. Attribute means column. Each column must have a unique name within that table (relation) and can be isolated across tables by qualifying the column name with the table name (such as *student.name)*. The domain specifies what are acceptable data values within a column (and by implication the acceptable operations on that column's data). The degree of a table is the width or number of columns and is part of the union compatibility criteria (the other one being that like-for-like columns in other results sets share comparable domains). The cardinality of a table is simply the depth or number of rows in the table. Good diagram gets bonus marks.

EXAMINER'S COMMENTS
Generally very well answered, with the vast majority of candidates obviously having clear understanding of TUPLE and ATTRIBUTE and most (but a smaller number) understanding DOMAIN, DEGREE AND CARDINALITY. Just about all candidates supplied a well-annotated sample relation. This was a strong question for the majority of candidates.
A common issue was to refer to DEGREE and CARDINALITY with respect to ER data models (one-to-many etc.) – something that was not expected for a question regarding *relational* concepts – but some marks were gained if the answer was well expressed. Extra marks were also awarded for well annotated diagrams of a relation.

(b)   *Using your own simple examples*, explain how the four SET operations work within Relational Algebra and what limitations the concept of UNION COMPATIBILITY places upon these sets being processed. Suitable Venn diagrams and sample relations will gain bonus marks.   **(10 marks)**

ANSWER POINTER

Extra marks are awarded for clear examples and diagrams but the key points are as follows.
UNION as the combination of all sets, INTERSECT as the 'overlap' of sets and MINUS as what is in one set but not another.
The role of duplicate removal from the final results set.
CARTESIAN PRODUCT (TIMES) should also be covered.
Suitable notation and clear use of Venn diagrams to highlight the universal set and the above three operators.
A clear definition of *union compatibility* - stating that in order to apply the set operators, all component sets must have the same degree (number of columns) and that comparable columns in each set must be of the same data type (compare like with like).
It should also be made clear that UNION (an operator to be applied) is not the same as UNION COMPATIBILITY (a condition to be satisfied).

EXAMINER'S COMMENTS
Not as popular as part (a) and often avoided by many candidates who had previously done well on part (a). Of those who did make an attempt, most did well but a common error was to discuss non-set theory relational operators like PROJECTION, SELECTION or JOIN. If this occurred but the answer was sound a sympathetic view was taken with the allocation of a few marks.
Pleasingly, unlike previous years, most candidates now know what UNION COMPATIBILITY is and how it differs from UNION. Oddly, very few covered CARTESIAN PRODUCT, focusing instead on the more famous UNION, INTERSECT AND MINUS. Overall, a poorly answered question for many CANDIDATES.

B5

EXAMINER'S GENERAL COMMENTS
The least popular question – but those who attempted it scored quite well.

(a)   Describe the various interfaces that a user may employ when interacting with a database, taking care to highlight the features, strengths and limitations of each. Your discussion should encompass all types of database user – from technical developers &

administrators to non-technical end-users. **(10 marks)**

ANSWER POINTER
This is quite an open type of sub-question that sets form-based interfaces into context. The better answer will include issues such as (non-technical) end-user interfaces like web-based (three-tier) interfaces, non-web (two-tier) GUIs such as forms & reports through to those interfaces designed for technical users like developers and DBAs such as forms/report generators and other software development environments – both graphical and command-line (for example Oracle's APEX and SQL*Plus respectively). It would also include third party interfaces for developers such as TOAD and specialized applications like Oracle's Enterprise Manager for DBAs.

EXAMINER'S COMMENTS
Not a particularly popular question and not particularly well done by those who attempted it. The most common issue was vague and rambling responses, rather than getting down to the different types of interface and the specifics of each. Many got side-tracked into discussing the 3-level ANSI-SPARC architecture rather than database interfaces.

(b) Explain what the term *data validation* means. *Using your own examples*, describe the various data validation techniques that may be embedded into a forms-based interface to a database. **(10 marks)**

ANSWER POINTER
The definition should be along the lines of 'ensuring that only clean, correct, accurate, well-formatted data is accepted into the database'. It should be about data integrity, not confused with database security (although validating a user ID and password is fine here). Typical examples will include format masks (for example, dates of birth or telephone numbers have valid structure), range checks (for example, financial fields are > 0.00), membership validity (for example, a supplier name is a valid entry in the database), cross-field consistency checks (so if the user enters 'Mr' in one field, the Sex field must be 'M' etc.), presence checks (to ensure no missed fields) etc. Clear examples needed for best marks.

EXAMINER'S COMMENTS
This sub-question invoked much better quality responses and those candidates who attempted it tended to do well. Most gave a clear overview of the key data validation issues, before moving on to cover specific ways that this could be achieved. Generally a well answered question.

(c) Describe the *form components* that may be used to implement these data validation techniques. **(5 marks)**

ANSWER POINTER
Form components would be drop-down lists to ensure only pre-validated entries can be chosen, radio buttons to ensure only a single (valid) option is selected, double-entry of key fields like passwords to rule out mistyping, automatic totaling of numerical data, on-form calendars where users can click on a given date, labels at side of each field with an example, on-form help button, highlighting which fields are mandatory via an asterisk etc.

EXAMINER'S COMMENTS
This part was universally well answered, with just about all candidates who attempted it providing detailed suggestions. Hence nearly all attempts were awarded the full 5 marks. By far the best answered part of this fifth question.

B6

EXAMINER'S OVERALL COMMENT

Nearly three quarters of the students attempted this question. Generally, high marks were awarded for what were generally a set of good answers.

(a) Explain what is meant by a transaction and why it is an important unit of operation in a DBMS? **(3 marks)**

ANSWER POINTER

Transaction: An action, or series of actions (e.g., update, insert…), carried out by a single user or application program, which reads or updates the contents of the database. A logical unit of work that transforms the database from one consistent state to another. (2 marks)

Its importance stems from the fact that it is the unit of concurrency and recovery control in a database system. (1 mark)

(b) Discuss the "ACID" properties of transactions. Give examples to illustrate your answer.
(12 marks)

ANSWER POINTER

For each: 2 marks for explanation + 1 mark for example

Atomicity The 'all or nothing' property. Students are expected to come up with an example of a transaction that includes a number of actions (e.g., many have used

the transfer of money between two bank accounts) and comment on how the transaction should not be left half-way through.

Consistency A transaction must transform the database from one consistent state to another consistent state, for example, making sure that all specified constraints (e.g., integrity and referential constraints) are enforced.

Isolation Transactions execute independently of one another. In other words, the partial effects of incomplete transactions should not be visible to other transactions. Students can show two concurrent transactions accessing the same data and comment on the mechanisms a DBMS can use to enforce isolation (locks, timestamps…).

Durability The effects of a successfully completed (committed) transaction are permanently recorded in the database and must not be lost because of a subsequent failure. Students can again use an example of a transaction that, for example, rolls back, and comment on how the DBMS handles such situation.

(c) Describe, with an example, one type of problem that can occur in a multi-user environment when concurrent access to the database is allowed.   **(6 marks)**

ANSWER POINTER

2 marks for description + 4 marks for an example such as Lost update problem, the uncommitted dependency problem, and the inconsistent analysis problem.

(d) Discuss how the log file is a fundamental feature in any database recovery mechanism.
                                                                                                        (4 marks)

ANSWER POINTER
Log files contains before and after-images of updates to the database. Before images can be used To undo changes to the database; after-images can be used to redo changes. Log records are written before the corresponding write to the database. Thus, if the system fails before the write occurs after the log record has been written but before the write to the database, the system can recover using the information in the log file. Log file also contains a checkpoint record, which can speed up the time for recovery following a failure.

EXAMINER'S COMMENT
Some candidates missed the fact that the question requested a discussion of the role of log files with regard to the DBMS recovery feature and not security. Hence, some answers were wrongly focused on how a log provides an audit trail of logins