

**BCS THE CHARTERED INSTITUTE FOR IT**  
**BCS HIGHER EDUCATION QUALIFICATIONS**  
**BCS Level 5 Diploma in IT**

**DATABASE SYSTEMS**

**EXAMINERS REPORT**

Thursday 1<sup>ST</sup> October 2015

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

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

Calculators are <b>NOT</b> allowed in this examination.
---

**Section A**

**A1**

**EXAMINER'S GENERAL COMMENTS**

This question was attempted by around 60% of all candidates with nearly half achieving a pass mark. The range of marks revealed a wide range of knowledge and ability. Overall the examiner perceived an improvement in the understanding of this key topic of data modelling. In this question candidates are required to apply data modelling knowledge and experience to a problem domain given in the form of a discourse. The initial question of starting the data model with a draft data model using a particular notation (UML) seemed to help candidates.

Refer to the following scenario and Figure A1 that follows.

'AZT' is a company that supply a number of **courses** (eg *Visual Basic; Python; SQL, Java*) delivered in the form of computer aided learning (CAL) **packages** that **students** study on-line. Students first register for an **attendance** on a course that AZT **offer** many times, occasionally over the same period of time. Then they **access** and work through a predefined set of packages in sequence (one at a time). After completing the last package for a course, students are then assessed by a practical test and the result is recorded as pass or fail. AZT log student access to each package recording the following information:-

*SessionID, The unique identifier whenever the package is accessed.*

*StudentID this is the identifier of the student logged into this session*

*PackageID this identifies the package a student is logged into*

*LoginTime this is the unique login time and date of this session*

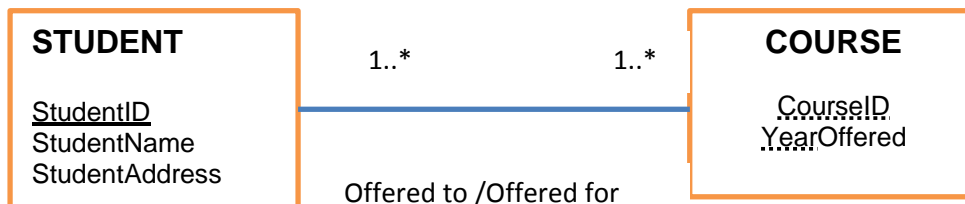
*LogoutTime this is the unique logout time and date of this session*

*LoginTimeLeft this is a derived field showing what time is left at the start of a session*

A student can access a particular package for a maximum of 18 hours over a predefined period after which the package is unavailable to that particular group of students. Each course has a selection of different packages unique to a particular course. The same course can be offered to students many times for example if they wish to re study.

An incomplete ER Model (using UML notation) has been produced below in Figure A1

**Figure A1 ER model (UML Class diagram notation) for use in question A1**



- a) Explain using the supplied ER model (figureA1) the concept of *cardinality* and *participation* constraints used to describe relationships between entity types.

**(5 marks)**

### Answer Pointer

Cardinality constrains how many instances (entities) participate in a relationship between two entity types. This qualifies the degree of a relation. So a 1 to many (represented as 1..\*) in the model can be qualified with participation. So one side may be obligatory making the constraint 1..\* or non-obligatory making the constraint 0..\*.

### EXAMINER'S COMMENTS

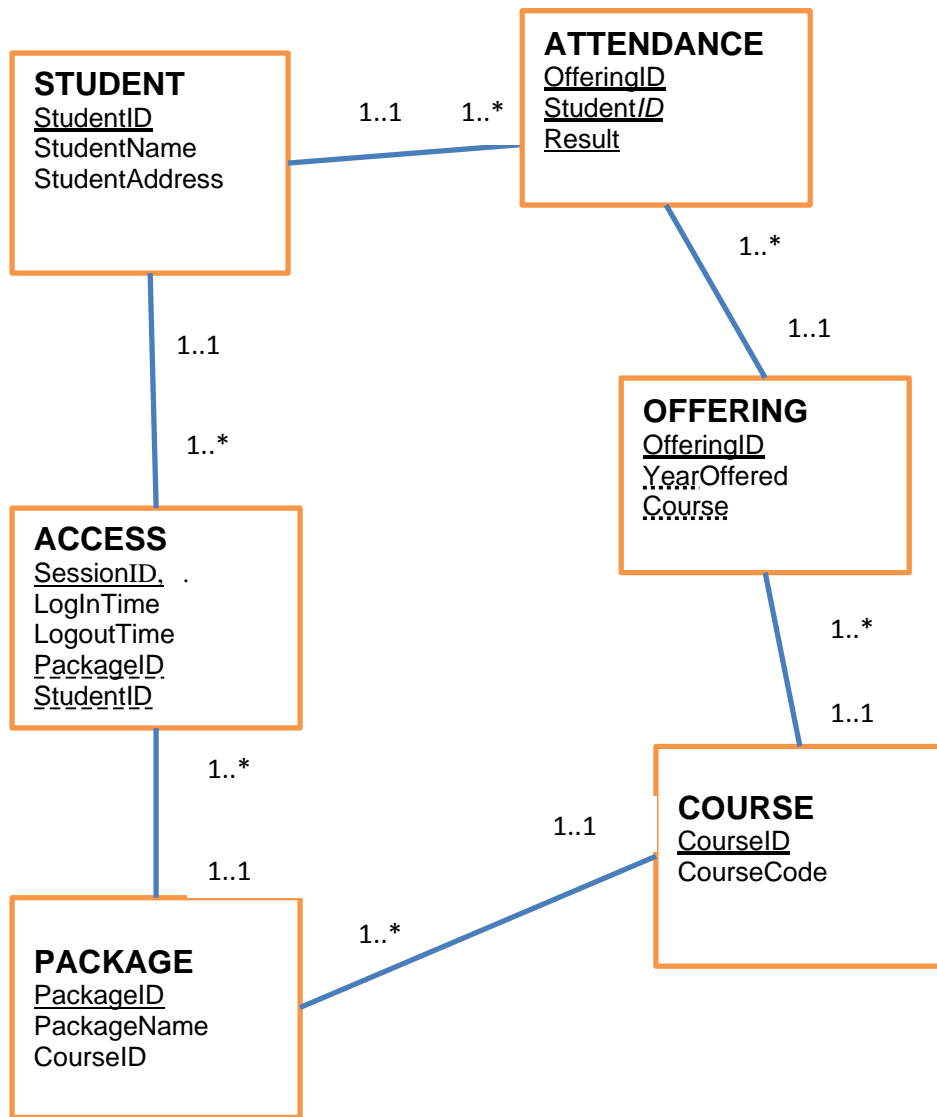
**This section was generally well answered with some misunderstanding of participation constraints.**

- b) Using any standard ER modelling notation, produce a **complete** ER model that includes **all** the Entity Types highlighted in bold in the scenario resolving any many to many relationships. State any assumptions you made.

**(14 marks)**

## Answer Pointer

FIGURE A15 ER Diagram (UML notation) SOLUTION part b



Comments on solution

Marks awarded for identifying Entity Types and correct placement of relationships.

Assignment of attributes. Naming of relationships not important

Attendance is a new Entity Type that reconciles the many to many relationship between Offering and Student.

Assumptions should not contradict the discourse

### **EXAMINER'S COMMENTS**

Many candidates made a good attempt at this ER modelling exercise. Marks were awarded for identifying correct relationships between individual pairs of entities. The answer provided is not definitive and many candidates produced creditable alternatives if they could justify assumptions that did not contradict the discourse. Many candidates failed to resolve many to many relationships perhaps because they did not read the question or were distracted by the supplied ER model that shows a many to many relationship between Course and Student.

c) Explain, using examples, the objectives of the following data modelling stages

Logical

Physical

**(6 marks)**

### **Answer Pointer**

An example might be derived from the scenario.

Logical

A logical ER model does not require a conceptual ER model, especially if the scope of the logical ER model includes only the development of a distinct information system. The logical ER model contains more detail than the conceptual ER model. In addition to master data entities, operational and transactional data entities are now defined. The details of each data entity are developed and the entity relationships between these data entities are established. The logical ER model is however developed independent of technology into which it is implemented.

Physical

One or more physical ER models may be developed from each logical ER model. The physical ER model is normally developed to be instantiated as a database. Therefore, each physical ER model must contain enough detail to produce a database. The physical model is normally instantiated in the structural metadata of a database management system as relational database objects such as database tables, database indexes such as unique key indexes, and database constraints such as a foreign key constraint or a commonality constraint. The ER model is also normally used to design modifications to the relational database objects and to maintain the structural metadata of the database. Physical data model will be influenced by the vendor specific implementation and the constraints that are imposed. Rules of normalisation still apply but the physical model forms the basis of optimisation in OLTP (i/o performance) vs OLAP (computation/query performance) are important. Marks allocated for examples from scenario etc.

### **EXAMINER'S COMMENTS**

There seemed to be quite a lot of confusion about what constitutes a logical data model. Although many candidates could relate to the theory of data independence, they could not always apply this in practice to reveal the different approaches necessary during these modelling stages.

A2

**EXAMINER'S GENERAL COMMENTS**

This question was attempted by around 54% of all candidates with nearly half achieving a pass mark. The range of marks revealed a wide range of knowledge and ability. In this question candidates were required to write SQL code to solve problems, and the examiner was happy that this question allowed candidates to sufficiently demonstrate a range of ability to solve problems using SQL code. Overall SQL knowledge is still rather patchy, however those with good familiarity with SQL had the opportunity to gain full marks. SQL is a topic that needs lots of practice and this question showed up much in the way of poor practical experience amongst many candidates.

Refer to the following Table definitions and then answer the question parts that follow.

Customer

Column Name	Type	Length	Nulls	Key
CustID	Integer	4	No	PK
CustName	Varchar	25	No	
Custaddress	Varchar	25	No	
CustPostCode	Char	8	No	

Products

Column Name	Type	Length	Nulls	Key
ProductID	Integer	4	No	PK
Description	Varchar	25	No	
StorageAddress	VarChar	25		
StoragePostCode	VarChar	12		
Reorderquantity	Integer	4	No	
QuantityInStock	Integer	4	No	
Unitprice	Decimal	8,2	No	
Total_quantitySold	Integer	4	Yes	

Orders

Column Name	Type	Length	Nulls	Key
OrderID	Integer	4	No	PK
CustID	Integer	4	No	FK
OrderDate	Date		No	
PromiseDate	Date		No	

LineItems

Column Name	Type	Length	Nulls	Key
ProductID	Integer	4	No	PK,FK
OrderNo	Integer	4	No	PK,FK
QuantityOrdered	Integer	4	No	

- a) With reference to the above table definitions explain the concept of a **Domain**. List 2 examples from the above tables of columns that could be defined as a domain.

**(5 marks)**

**Answer pointer part a)**

Domains represent a set of data with specific properties related to a column in a table but independent of a relation. Whereas columns (or attributes) are associated and dependent on a relation. A suitable domain would be Address and post code as these have common properties and constraints. Dates are another domain candidate

Domains are not supported by every DBMS vendor

*Example SQL domains according to SQL standard*

*CREATE DOMAIN DDate as*

*date*

*default current\_date*

*not null*

*CREATE TABLE ( t1 sid varchar 50), DDate ) an attribute associated with a table and a the domain defined above*

**EXAMINERS COMMENTS**

**This question was reasonably well answered in most cases. However, many candidates' answers were verbose and lacked examples. For example, it would have been beneficial to include some example SQL code that creates and applies DOMAINS.**

- b) Suppose you were required to implement (create and populate) the tables listed above using an SQL script containing a set of CREATE TABLE and INSERT operations. List the logical sequence the tables should be populated to ensure referential integrity is preserved.

**(4 marks)**

**Answer Pointer**

Products, Customers, Order (depends on Customer), LineItems (depends on Product and Order)

**EXAMINER.S COMMENTS**

**There was some misunderstanding amongst candidates. This is a common problem when writing a SQL script to create a database schema that reflects logical order and to apply referential integrity. Despite an apparent lack of practice in writing SQL scripts, there were many correct answers. Sometimes candidates wrote the full SQL CREATE script rather than a simply a LIST of the tables. Candidates should read the question carefully. The order could be either Products or Customers as the first batch of INSERT statements followed by the referenced tables. Also some common sense thinking is required to establish that an Order cannot exist without an existing Product.**

- c) TotalQtySold is a column that contains derived data. Write a SQL SELECT statement that computes a value for this column for a specific product.

**(5 marks)**

**ANSWER**

```
SELECT p.quantityInstock - li.itemsordered AS totQtysold
FROM products p, lineitems li
WHERE p.productID = li.productID
AND p.productID = 3
```

**EXAMINER'S COMMENTS**

A reasonable number of correct answers were submitted but many candidates omitted the important SUM operator. The examiner noted the column was named slightly differently as TotQtyOrdered. This didn't seem to have any effect on candidates' answers.

d) What are the advantages and disadvantages of storing derivable data in a table?

(4 marks)

**Answer Pointer**

The data needs updating automatically for every transaction on the affected tables otherwise becomes out of date. Overhead of computation could be severe for intensive OLTP apps.

**EXAMINER'S COMMENTS**

This section was generally well answered

e) Write an SQL UPDATE statement that updates the column TotalQtySold given a specific product.

(7 marks)

**Answer Pointer**

```
UPDATE p
SET p.tot-Qtysold = p.QuantityInStock - li.itemsOrdered
FROM products p, lineitems li
WHERE p.productID = li.productID
AND p.productID;
```

**EXAMINER'S COMMENTS**

Most candidates had difficulty in formulating SQL Update code, resulting in very few correct answers. Some marks were awarded for attempts where candidates could identify the correct structure of UPDATE statements. For example SET clause ; SELECT clause, WHERE condition clause. (The code from part (c) is part of the UPDATE).

A3

**EXAMINER'S GENERAL COMMENT**

This was an extremely popular question with almost all candidates attempting it. The question had a high pass rate with many candidates producing excellent attempts.

(a) A University stores student records in the format shown below:

<b>StudentID:</b> S0012	<b>Name:</b> John Silver	<b>Course:</b> Computing	
<b>Module Code</b>	<b>Module Name</b>	<b>Grade</b>	<b>Pass/Fail</b>
M001	Databases	C	Pass
M002	Web Technologies	E	Fail

Module grades A-D are pass grades and E-F are fail grades.

- (i) Identify the repeating group of attributes and transform the above format into tables that are in 1<sup>st</sup> Normal Form. **(4 marks)**
  
- (ii) Identify any partial dependencies and transform into tables that are in 2<sup>nd</sup> Normal Form. **(5 marks)**
  
- (iii) Identify any transitive dependencies and transform into tables that are in 3<sup>rd</sup> Normal Form. **(2 marks)**

(a)

**Answer Pointer**

- (i) (ModuleNo, ModuleName, Grade, PassFail) is the repeating group. (1 mark)

We split the table into two 1NF tables.

Student (StudentID, Name, Course) **(1 mark)**

Result (StudentID\*, ModuleNo, ModuleName, Grade, PassFail) **(2 marks)**

- (ii) Partial dependency: ModuleNo → ModuleName (2 marks)  
We remove this partial dependency to a separate table.

Student (StudentID, Name, Course)

Result (StudentID\*, ModuleNo, Grade, PassFail) **(1 mark)**

Module (ModuleNo, ModuleName) **(2 marks)**



(iii) Transitive dependency: Grade  $\rightarrow$  PassFail (1 mark)  
Result (StudentID\*, ModuleNo, Grade)

Grade (Grade, PassFail) (1 mark)

Alternatively, students were also given the 2 marks if they did not consider it appropriate to normalise at this stage. For example, instead of creating a new table, a derived value for PassFail could be calculated.

### EXAMINER'S COMMENTS

**Most students managed to correctly normalise the given scenario.**

(b) (i) Explain the term *candidate key*.

(2 marks)

(ii) List **three** candidate keys for the following table (A, B, C and D are the attributes of the table):

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
a1	b1	c1	d1
a2	b3	c3	d1
a1	b2	c1	d2

(6 marks)

### Answer Pointer

(i)

A candidate key is a minimal set of attributes that uniquely identifies each row in a table

(2 marks)

(ii)

(2 marks each)

(B), (A, D), (C, D)

### EXAMINER'S COMMENTS

**Most candidates managed to give a reasonably accurate definition of a candidate key. However, most fail to note that it has to be minimal and this had an impact on answering part (ii). For example, once it was recognised that "B" can be a candidate key, any combination of columns involving "B" again (e.g., "BC") cannot be a candidate key. Other candidates struggled with the fact that abstract values (a1, a2...) were used in the scenario instead of concrete values. This was deliberately used in order to test the candidates' understanding of a candidate key.**

- (c) The following table stores details of employees and the projects they work on and for how long. The Primary Key is (EmpID, ProjID).

**Workload**

<u>EmpID</u>	EmpName	<u>ProjID</u>	ProjName	HoursPerWeek
E01	Smith	P02	Database	10
E01	Smith	P01	Web Portal	5
E02	Robinson	P02	Database	20

- (i) Explain why the above table is not in 2<sup>nd</sup> Normal Form.

**(2 marks)**

- (ii) Transform the table into 2<sup>nd</sup> Normal Form tables.

**(4 marks)**

**Answer Pointer**

- (i)

There are two partial dependencies (2 marks):

EmpID → EmpName

ProjID → ProjName

- (ii)

Employees (EmpID, EmpName)

**(1 mark)**

Projects (ProjID, ProjName)

**(1 mark)**

WorkLoad (EmpID\*, ProjID\*, HoursPerWeek)

**(2 marks)**

**EXAMINER'S COMMENTS**

Many students recognised the partial dependencies and did a good job of normalising the given table. However, there were many others who could not work out the proper normalisation. This is perhaps due to the fact that there are two, instead of one, partial dependencies.

## Section B

B4

### EXAMINER'S GENERAL COMMENTS

This was a very popular question. Those candidates who knew the material scored heavily with well over half achieving pass marks, but there was a wide variation in the standard of attempts.

(a) *Using your own simple Relational Algebra (RA) examples and any appropriate diagrams*, list the eight RA operators, explaining the essence of each and providing at least one suitable example based on *a sample relation of your own choosing*. Good diagrams will gain extra credit. You should also highlight which RA operators originate from mathematics and which were developed specifically for relational databases – and by whom. **(10 marks)**

### Answer Pointer

This will be marked holistically with extra marks for good examples and clear diagrams but good students should cover: SELECTION, PROJECTION, JOIN, DIVIDE, CARTESIAN PRODUCT (TIMES), UNION (and maybe UNION ALL), INTERSECTION AND MINUS (DIFFERENCE) operators. The first four were developed specifically for relational databases by Dr. Ted Codd while the rest originate from mathematics (set theory). For each, a Venn diagram and suitable RA example is expected.

### EXAMINER'S COMMENTS

There was a huge spread of quality in this section. Some candidates practically wrote out a textbook and thus got full marks while others either skipped it completely or offered the most flimsy (and often confused) answers. Many candidates named the operators correctly but then either confused them with each other or failed to provide a concrete example.

(b) For each of the following two relational concepts, explain the key ideas behind them and, *using a sample relation of your own choosing*, provide suitable examples...

- Entity Integrity
- Referential Integrity

Each item is worth five marks

**(10 marks)**

Entity Integrity – all about primary keys (following concepts + examples):

- No nulls
- No duplication
- Must guarantee unique row identification – forever
- Atomic versus composite candidate and primary keys

Referential Integrity – all about foreign keys (following concepts + examples):

- FK column(s) must match domain of referenced column(s)
- Names do not matter
- Can contain nulls
- Can contain duplicates
- ON DELETE CASCADE etc

#### **EXAMINER'S COMMENTS**

The consistency in responses was better here but still there was evidence of substantial variation in appreciation of these core concepts. Most candidates picked up on the fact that the question related to primary and foreign keys but they then failed (in many cases) to give a full account of all the issues outlined in the above marking scheme. A small number of students again gave textbook answers and were suitably rewarded.

(c) Write a *single sentence* with a *simple example* (based on *any sample* relation) to illustrate the following relational concepts...

- Candidate Key
- Alternate Key
- Atomic Key
- Composite Key
- Primary Key

Each item is worth one mark

**(5 marks)**

#### **Answer Pointer**

Any sensible comments accepted as long as the key concept(s) are captured. Only award full marks if some concrete examples are also provided.

#### **EXAMINER'S COMMENTS**

Apart from the relatively small, number of students who gave textbook answers, there was a surprising level of confusion and vagueness regarding these concepts. While almost everyone gave a good account of 'primary key', the other concepts are not well known to many others.

## B5

### EXAMINER'S GENERAL COMMENTS

This was the least popular question with less than one third of candidates attempting it. However, the pass rate was well over 50%.

(a) Using a *suitable diagram and any appropriate examples*, address the following two points:

- Describe how *database forms* relate to the three-level ANSI-SPARC architecture of a typical database system
- Describe how *SQL views* relate to the three-level ANSI-SPARC architecture of a typical database system and how they compare with database forms

**(10 marks)**

### Answer Pointer

This question is essentially about how a form relates to the external level of the ANSI-SPARC model as a means of providing one (of possibly many different) user views onto the underlying conceptual level and the database it logically describes. A good diagram is essential. The better candidates will then go on to discuss SQL views as a programmatic equivalent of a form in the sense that the defining logic of the view provides the same data to the end user – albeit not in a graphical way. Good SQL code examples needed for higher marks.

### EXAMINERS COMMENTS

**The ANSI-SPARC work was well explained and, in the bulk of cases, supported by a good quality, well-annotated diagram. The role of views within this framework was less well covered with many candidates simply ignoring this second aspect of the question**

(b) *Data validation* is a key requirement when entering data into a database. Discuss the relative strengths and weaknesses of performing this data validation at the *application form* level and at the *database level* as well as briefly explaining the methods that each level uses to ensure that validation is achieved.

**(10 marks)**

### Answer Pointer

At the form level (front-end) the data can be validated using on-form logic and by avoiding invalid data in the first place by using 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. - thus avoiding sending erroneous data to the back-end database and wasting a return journey over the network. So form level validation is simple but efficient.

At the database level (back-end), much more complex data validation – that requires access to the database tables or the data dictionary perhaps - can be employed. This necessitates a round trip to the server but it does mean that complex business logic (written as table constraints or SQL triggers) can be enforced. So database level validation is vastly more powerful but slightly less efficient.

**EXAMINER'S COMMENT:**

**Many Good answers were submitted, with the vast majority of candidates showing an understanding of all the core ideas.**

(c) Explain the roles, responsibilities and relationships of application forms and databases in a three-tier web-based architecture, taking special care to discuss the concepts of presentation, business logic and data management. You are not expected to write any software code but you should support your discussion with a clearly annotated diagram illustrating how all three components of a three-tier architecture interact and where they reside within that structure.

**(5 marks)**

**Answer Pointer**

The front-end web form, running inside a web browser (client), is responsible for data presentation and some simple data validation, the back-end DBMS is responsible for all data management tasks and for input/output of data to/from the database while the 'middleware' is the web server that acts as the dynamic glue between the front-end user form and the back-end database by providing the business application logic that both constructs queries to interrogate the database (based on user input) and then constructs in real-time a webpage of data results to be sent back to the web browser. A good diagram is essential for full marks.

**EXAMINER'S COMMENTS**

**Performance was patchy on this question, ranging from excellent – both narrative and diagrams – to outright omission – and every shade of response in between.**

**B6**

**EXAMINERS' GENERAL COMMENTS**

**Two thirds of the candidates attempted this question. There was a high pass rate showing good familiarity with this topic.**

A major objective of the ANSI-SPARC architecture is to provide data independence.

(i) Draw a diagram illustrating this architecture. **(3 marks)**

(ii) Using examples, discuss the concepts of logical data independence and physical data independence. **(6 marks)**

### Answer Pointer

(i) Diagram illustrating the external, conceptual, and internal levels. **(3 marks)**

(ii)

Logical data independence: the immunity of the external schema to changes in the conceptual schema. For example, the addition or removal of entities, attributes... should be possible without having to change the external schema or re-write the application programs. **(3 marks)**

Physical data independence: the immunity of the conceptual schema to changes in the internal schema. For example, changes to the file organisation or storage devices... should be possible without having to change the conceptual or external schemas. **(3 marks)**

### EXAMINER'S COMMENTS

**Most candidates managed to draw a proper diagram involving all three levels. However, the explanation provided did not always show a good understanding of how data independence works.**

a) Describe four features (functions) you would expect to find in a DBMS. **(8 marks)**

### Answer Pointer

The description of any sensible four features, such as the following: (2 marks each)

Data storage, retrieval and update (using SQL...)

Transaction support

Security (encryption, authentication and authorisation...)

Concurrency control services for multiple users

Backup and Recovery services

Integrity services

...

### EXAMINER'S COMMENTS

**Again, most students did a good job of describing some of the important features of a DBMS. The majority of the answers listed: Security, Concurrency Control, Backup and Recovery, and Data storage and retrieval.**

b) The 'client-server architecture' is commonly used to implement a database system.

(i) Draw a diagram to illustrate this architecture. **(3 marks)**

(ii) Describe the advantages of this approach and comment on whether it is appropriate for the Web. **(5 marks)**

### **Answer Pointer**

(i)

Diagram showing the three tiers architecture: client, application server, and database server

**(3 marks)**

(ii)

Advantages include, for example:

**(3 marks)**

- Better performance and consistency
- Likely reduction in hardware costs
- Reduction in communication costs

Architecture maps quite naturally to the Web with a Web browser acting as first tier, a Web server/application server acting as second tier and database server as third tier. **(2 marks)**

### **EXAMINER'S COMMENTS**

**All candidates managed to draw a reasonably good diagram. However, not many provided a good explanation of the advantages of the architecture or how it maps to the Web.**