BCS THE CHARTERED INSTITUTE FOR IT

BCS HIGHER EDUCATION QUALIFICATIONS
BCS Level 5 Diploma in IT

DATABASE SYSTEMS

Thursday 29th September 2016 - Morning
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.

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

SECTION A

Answer Section A questions in Answer Book A

A1

Examiner's General Comments

Around half of the candidates made an attempt at this question and 50% of the attempts achieved pass level with some scoring maximum marks. The performance overall was a little bit lower than expected on this question.

a) One of the main responsibilities of a database developer is to enforce the following integrity constraints on database tables :-

Domain Integrity

Entity Integrity

Referential Integrity

i) Write SQL code to show how you would implement *EACH* of these constraints on the Tables provided in Figures A1.1 to A1.3 below.

(12 marks)

ANSWER POINTER

An indication of understanding these terms in the context of SQL gains credit. Marks are also awarded for good code structure allowing for minor syntax errors :-

Domain Integrity : Restricts data ranges and imposes datatyping. There should also be evidence of legislation for NULL values.
Entity Integrity : Restrict duplication ensuring entity identifier behaves as a unique/key
Referential Integrity : Ensure matching values between primary key and foreign keys

Credit is given for examples of how each of these constraints are specified within SQL `CREATE TABLE` statements for each of the tables (FigA1.1 to FigA1.3) listed below.

```
CREATE TABLE Hotel_Package (HotelCode CHAR(3)
                           ,PackageID INT
                           ,Price DECIMAL(5,2) NOT NULL
                           ,CHECK (Price > 0)
CONSTRAINT c_pk PRIMARY Key (HotelCode,PackageID)
CONSTRAINT cp1_fk FOREIGN Key HotelCode REFERENCES Hotel.HotelCode
CONSTRAINT cp2_fk FOREIGN KEY PackageID REFERENCES Package.PackageID
);

CREATE TABLE hotel (HotelCode CHAR(3)
                   ,Hotel VARCHAR(50)
                   ,Resort VARCHAR(23) NOT NULL
CONSTRAINT h_pk PRIMARY KEY (HotelCode))

CREATE TABLE Package(PackageID CHAR(2) NOT NULL
                      ,Catering CHAR(2) NOT NULL
                      ,Nights int CHECK (Nights > 0)
                      ,Months VARCHAR(10) NOT NULL
CONSTRAINT pack_pk PRIMARY Key (PackageID))
```

ii) Write TWO SQL statements, one an INSERT, the other a DELETE statement, that will test whether your referential integrity constraints are actioned.

(4 marks)

ANSWER POINTER

For example 2 marks each statement

INSERT a row into the `Hotel_Package` table.- the statement suffers the constraint that the values of HotelCode must exist in the `Hotel` table and values of `PackageID` must exist in the `Package` table

```
INSERT INTO hotel_package VALUES(‘BBC’,9,1972);
```

DELETE - cannot delete a row containing a HotelCode value referenced in another table

```
DELETE FROM hotel WHERE resort = ‘Benidorm’;
```

b) One of the main responsibilities of a DBA (database administrator) is to enforce security measures on user access to database data.
Describe these security measures and provide examples, written in SQL, of their construction, using the sample tables provided.

(9 marks)

ANSWER POINTER

Access Control using passwords;

Restricting data visibility;

Creating roles to act as a privilege management tool, data dictionary/meta data, create profiles to restrict access

VIEWS ;-

For example, to restrict data visibility use `VIEWS`

```
CREATE VIEW V1 AS (

SELECT HotelCode , PackageID

FROM Hotel_Package

WHERE HotelCode  = ‘CBR’;
```

GRANT

Apply various types of privileges using SQL `GRANT` such as

Delete data from a specific table.

Insert data into a specific table.

Create a foreign key reference to the named table or to a subset of columns in a table.

Select data from a table, view, or a subset of columns in a table.

Update data in a table or in a subset of columns in a table.

Examples might be

To allow `SELECT` access on table `Hotel` to the authorised users mary and harry:

```
GRANT SELECT ON TABLE Hotel TO mary,harry
```

To grant the `SELECT` privilege on table Hotel to the `manager` role:

```
GRANT SELECT ON TABLE Hotel TO manager_role;
```

Apply your example SQL code to the tables given in Fig A1.1 Fig A1.2 and Fig A1.3 below.

Fig A1.1 HOTEL Table

HOTEL_CODE	HOTEL	RESORT
FLB	Flamingo	Benidorm
BHB	Bali Hai	Benidorm
HAZ	Hawaii	Santa Posa
SPZ	Sun Park	Playa Blanca
AHB	Al Hambra	Benidorm
JDM	Jardin del Sol	Palma Nova
SPB	Sun Park	Playa Blanca

Fig A1.2 HOTEL_PACKAGE Table

HOTEL_CODE	PACKAGE_ID	PRICE
FLB	1	265
JDM	1	295
BHB	3	199
HAZ	4	308
SPZ	6	310
AHB	2	199
JDM	3	199
JDM	6	169
SPB	6	159

Fig A1.3 PACKAGE Table

PACKAGE_ID	CATERING	NIGHTS	MONTH
1	SC	7	June
3	FB	14	November
4	HB	10	July
6	FB	10	November
2	HB	14	May

Examiner Comments

Although there was evidence of good knowledge of SQL in parts b) and c) there were also some poor attempts showing a lack of practice in writing SQL. Candidates are advised with this type of question to gain some practical experience in SQL using examples from text books or web sites. Surprisingly most candidates omitted check constraints as a method of defining domain constraints. With regards to checking referential integrity in part a (ii) candidates should note that two different tables are required - one suffering an insert operation and the other being affected by the foreign key reference preventing a delete. In part b) there was reasonable knowledge of VIEWS and GRANTS along with concepts of authorisation and permissions. To attain full marks, reasonably accurate SQL code applied to the example tables rather than textbook answers was required.

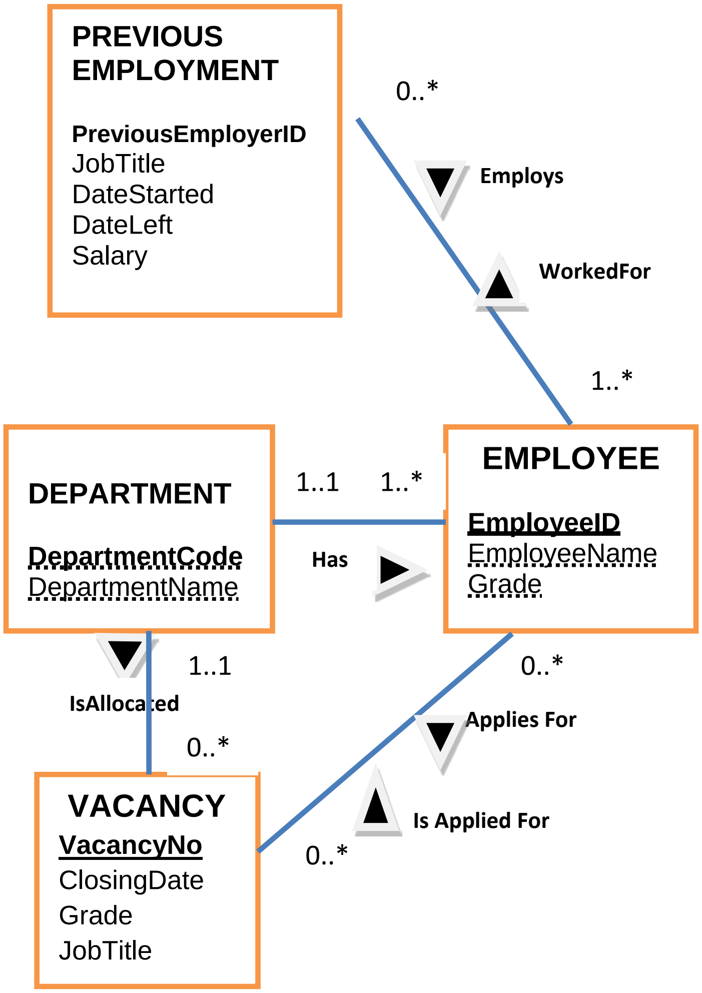
A2

Examiner's General Comments

Around 60% of candidates attempted this question. The pass rate was a fairly low 37% perhaps reflecting the challenge of an abstract topic like data modelling.

Refer to Fig A2.1 below, this is an Entity relationship (ER) data model presented in UML notation used in a Human Resources (HR) database for a Company.

Figure A2.1 HR data model for use in question 2



a) Explain the benefits of using ER data modelling techniques to assist in the design of a relational database.

(6 marks)

ANSWER POINTER

Top down approach is usually necessary when the target database is quite large with many entities

Easy to check access paths to the data and confirm data requirements can be met.

Break down/separate the stages of database design from analysis (conceptual) to physical modelling.

Easier to visualise for customers making it more natural to interact and get involved as a data model deals with real world features/characteristics (entities/objects/classes) within a client's domain of discourse.

Visible modelling of constraints which can map directly to Tables in at least 2NF

A data model offers an alternative route to normalisation/FD (the bottom-up approach) ideally in combination.

b) Identify the THREE types of data modelling constructs used in the ER data model given in Fig A2.1 and state how they influence the design of relational database tables.

(6 marks)

ANSWER POINTER

The main modelling constructs are

- Entity Types,
- Relationship types,
- Relationships - degree and participation constraints
- Attribute types

They influence the design of a relational database directly by mapping entity types to tables; assign attributes if many to many relationships have been resolved. Relationship types might indicate placement of foreign keys and indicate referential integrity. Relationship degree and participation can influence the number of tables to avoid unnecessary use of NULL values. An ERD maps entity types normally resulting in a design that conforms to 3NF

c) The data model given in Fig A2.1 has omitted a key step in logical database design. The omission of this step means that logical entities cannot be directly mapped to physical tables. Explain what key step is missing and give ONE example that shows how you would update the data model given in Fig A2.1 to resolve this omission.

(6 marks)

ANSWER POINTER

The many to many relationships need to be resolved. One of them needs to be chosen for discussion - for example Employee-> previous company.

The change from a many to many relationship involving two entities will result in a third intersection entity. In the case of the Employee to Previous_company relationship, the new entity would act as a linking entity to support the employment history of the employees. Note how the relationship cardinality is reflected in this change.

d) Explain how you would extend the data model given in Fig A2.1 to represent the following requirement.

Candidates who apply for a vacancy that a department wants to fill are employees of the company. A vacancy is normally filled following one or more ***interviews****. In an interview each candidate (the interviewee) is interviewed by another employee (the interviewer). Following an interview a vacancy may or may not be filled. If it is a successful candidate is appointed to a new* ***position*** *within the company. Unsuccessful candidates continue in their current* ***position****.*

The database needs to record information about the interviews conducted for a particular vacancy. Who is interviewed, by whom and the outcome (offer of a position or a rejection). The new ***position*** *that an employee is appointed to must also be recorded. This includes the unique Position Number, the Job Title, the salary and the start date.*

Please Note: You must state any assumptions you made and make sure you adhere to a standard modelling notation.

(7 marks)

ANSWER POINTER

There is no definitive solution for this part of the question. It tests the skill of the candidate in analysing a data model and applying modelling decisions. For this reason, any necessary assumptions must be declared and any modelling decisions clearly stated. In addition candidates should state how their model meets the specified requirements.

The indicative solution deals with the requirements in the discourse as follows::

Requirement: *“record information about the interviews conducted for a particular vacancy”.*

Modelling decision :-

The new entity type **Interview** (emphasised in the discourse) is introduced and has a relationship to **Application** (another new entity type created between Vacancy and Employee) to resolve the Many to Many relationship.

Assumption: a particular application for a vacancy ***may*** result in normally one (or more) Interview(s) but an Interview **must** be associated to one application.

Requirement: *“Who is interviewed, by whom and the outcome (offer of a position or a rejection)”*

Modelling decision :-

There is a need for two relationships; one between Application and Interview and one between Application and Employee. This models the Employee roles of Interviewer and Interviewee. The Interviewee is related to their application, whereas the Interviewer is related to the Interview they conduct.

Assumption: not applicable as this is stated in the discourse.

Requirement:*The new* ***position*** *that an employee is appointed to must also be recorded. This includes the unique Position Number, the Job Title, the salary and the start date.*

Modelling decision :-

A new Entity Type Position is introduced related to Employee.

Examiner's Comments

Many candidates seem to find data modelling a challenge and struggled with this question, particularly the main modelling part (part d). This perceived weakness could be addressed, by carefully reading the scenario/requirements and undertaking plenty of data modelling practice.

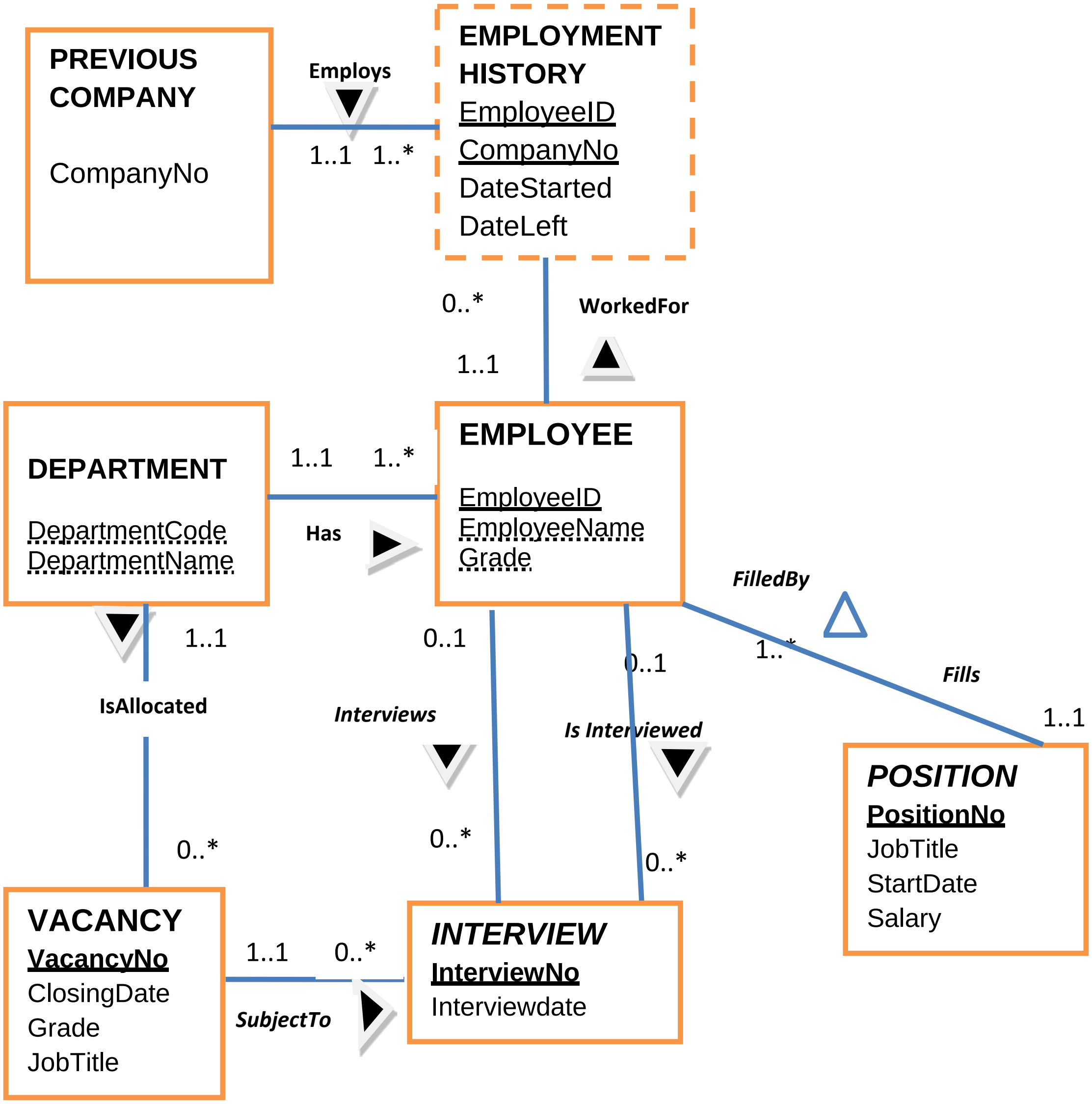
Part a) was answered reasonably well with some good answers. However many candidates produced stock answers with little reflection.

In part b) about a quarter of candidates interpreted this as meaning the three variations of relationships, that is, 1:1; 1:M and M:M and although this is part of the answer these candidates thought this was the full answer. Some other candidates incorrectly answered the question by dealing with entity integrity and referential integrity.

Part c) was fairly well answered with either example presented correctly by most candidates. But again a few candidates presented incorrect solutions; such as stating foreign or primary keys were missing in the ERD.

In part d), there were many unexpected different interpretations of the scenario. Very few candidates picked up on the requirement to model interview/Application – employee relationships. Sometimes it could be seen how the candidate arrived at a solution, but often it was unclear how they had reasonably addressed the requirements. In all cases, candidates are advised to clearly state any assumptions they make and take care not to contradict the discourse. Also care must be taken in the accuracy of cardinality constraints expressed in UML notation.

Figure A2.1S data model solution part c) dotted and part d) Italics extra E/R in question



A3

Examiner's General Comments

This was a very popular question with almost all candidates attempting it. The pass rate of the question was 72.20%, with an average mark of 12.66 and a standard deviation of 5.68.

(a) A business keeps invoices in the format shown below:

<table>
<tr><td colspan="4"><b>customerID:</b> C12 <b>custName:</b> John Silver <b>custAddress:</b> 47 High Street, London</td></tr>
<tr><td><b>productCode</b></td><td><b>prodName</b></td><td><b>Price</b></td><td><b>Quantity</b></td></tr>
<tr><td>P1</td><td>Laptop</td><td>£300</td><td>2</td></tr>
<tr><td>P2</td><td>iPad</td><td>£250</td><td>3</td></tr>
</table>

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

(4 marks)

ANSWER POINTER

(productCode, prodName, price, quantity) is the repeating group.

The table should be constructed with a key of customerID and productCode

Invoice (customerID, productCode, custName, custAddress, prodName, price, quantity)

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

(5 marks)

ANSWER POINTER

Partial dependency: productCode → prodName, price
Partial dependency customerID → custName, custAddress

Partial dependencies need to be removed by creating separate tables.
Customer (customerID ,custName,cust Address)

Product (productCode, prodName, price)

Invoice (customerID, productCode, quantity)

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

(2 marks)

ANSWER POINTER

There are no transitive dependencies, so the tables are in 3NF.

Examiner’s Comment

Most candidates managed to correctly normalise the given scenario. In general, candidates were able to deal with partial and transitive dependencies appropriately

(b) An embassy records details of interviews of visa applicants in the table

Below. Interviews are conducted by members of staff in some of the embassy rooms. In any given day, a member of staff tends to use the same room throughout that day. An applicant cannot have two interviews in the same day.

applicantNo	**interviewDate**	**interviewTime**	**staffNo**	**roomNo**
AP1	13-May-2016	10:30	S5	R101
AP2	13-May-2016	12:00	S5	R101
AP3	13-May-2016	12:00	S9	R200
AP2	22-Sep-2016	10:30	S5	R200

(i) Explain the term *candidate key*.

(2 marks)

ANSWER POINTER

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

(ii) List **three** candidate keys for the above.

(6 marks)

ANSWER POINTER

(applicantNo, interviewDate),
(staffNo, interviewDate, interviewTime),
(roomNo, interviewDate, interviewTime).

Examiner’s Comment

Only a few candidates attained full marks for this question and the majority failed to recognise the need for composite candidate keys. It is important that candidates are able to put the definition into practice and look for columns that yield unique combinations of values.

(c) The following table stores details of doctors, patients and dates of appointments. The Primary Key is (doctorID, patientID).

Appointments

doctorID	**doctorName**	**patientID**	**patientName**	**date**
D01	Kumar	P02	Smith	10-Aug-16
D01	Kumar	P01	Ford	5-Sep-16
D02	Robinson	P02	Smith	10-Aug-16

(i) Explain why the above table is not in 2nd Normal Form.

(2 marks)

ANSWER POINTER

There are two partial dependencies.
doctorID → doctorName
patientID → patientName

(ii) Transform the table into 2nd Normal Form tables.

(4 marks)

ANSWER POINTER

Doctors (doctorID, doctorName)
Patients (patientID, patientName)
Appointments (doctorID*, patientID*, date)

Examiner's Comment

Almost all candidates managed to identify the partial dependencies, but disturbingly, quite a few did not get the transformation right.

B4

Examiner's General Comments

This was the least popular question with 26% of the candidates making an attempt. The pass rate was lower than expected given that the material covered by this question is of fundamental importance in database technology.

(a) Using your own simple examples and any diagrams you feel suitable, explain how the concept of *constraints* can be enforced within relational theory.

(10 Marks)

ANSWER POINTER

A good response would address the following:

The role of domains in constraining the range of values allowed in columns (attributes).

The use of primary keys to enforce entity integrity (no nulls and no duplication).

The use of foreign keys to enforce referential integrity rules.

The use of UNIQUE (alternate key) and NOT NULL constraints

Credit could also be given for CHECK constraints as applicable.

Examiner's comment

Not a popular question but on attempting it candidates scored well. A few candidates discussed ER model constraints – such as one-to-one, but as the question specifically asked about the relational model, such responses attracted no marks.

(b) Using your own simple examples and any diagrams you feel suitable, explain how the concept of *structure* can be demonstrated within relational theory. You should clearly illustrate the key terminology involved.

(15 Marks)

ANSWER POINTER

A good response will address the following points.

The relation as a tabular storage concept made up of rows and columns – the fundamental concept and unit of storage.

Each relation to have a unique name.

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 the range of acceptable data values within a column (and by implication the acceptable operations on that 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 result sets share comparable domains).

The cardinality of a table is simply the depth or number of rows in the table.

A good diagram attracts marks.

Primary key should mention uniqueness criteria and absence of NULL entries and its role in identifying each tuple plus possibly single column versus composite PK.

Foreign key should mention that it represents a relationship between two tables/relations and that duplication and NULLs are allowed in the foreign key column, but if a value exists it must map onto an existing value in the other table's primary (strictly candidate) key. Credit could also be gained by mentioning restriction of changes to the primary key in the parent relation and the option for cascade delete if a parent row needs to be deleted.

Examiner's comment

Again, not a popular question but when this section was attempted it was well done.

B5

Examiner's General Comments

A fairly popular question with 67% of candidates submitting an attempt. Two thirds of the attempts achieved pass level and some exceptionally good answers were received.

(a) There are many ways in which a user can interface to a database. Using your own simple examples and any diagrams you feel suitable, describe the key features, strengths, weaknesses and typical uses & users of the following types of interface:

i) Text-based

(5 Marks)

ii) Form-based

(5 Marks)

iii) Web-based

(5 Marks)

ANSWER POINTER

This is quite an open type of sub-question that sets interfaces and databases into context. The better students 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 also includes third party interfaces for developers such as TOAD and specialized applications like Oracle's Enterprise Manager for DBAs. It would also be nice to see comments regarding interactive use versus scripted or programmatic usage.

Examiner's Comment

Generally poorly attempted with many vague and rambling responses that only rarely touched upon the issues highlighted in the marking scheme.

(b) End-users have a tendency to make mistakes. While the database will (hopefully) have a wealth of data integrity constraints to prevent erroneous data getting into the database, it is desirable to stop such bad data ever getting to the back-end database in the first place. Describe the *user-interface* components and techniques - that appear on forms and websites that may be used to implement these data validation techniques. Why is it preferable to catch invalid data at the interface level rather than at the database level?

(10 Marks)

ANSWER POINTER

Form and web 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 totalling of numerical data, on-form calendars and date pickers 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.

It is preferable to catch such problems at the interface as this avoids transmitting flawed data over the network, thus minimizing wasted network traffic, avoiding unnecessary DBMS processing, avoiding transmitting potentially dangerous code/data to the database (as in SQL injection) as well as the obvious time saved.

Examiner's comment

Mostly well answered with the vast majority of attempts doing a good job in explaining the issues covered in the marking scheme.

B6

Examiner's General Comment

This was a popular question, with 80.58% of candidates attempting it. The pass rate was 71.08%, with an average mark of 13.07 and a standard deviation of 6.12.

a) The ANSI-SPARC architecture provides data independence.

(i) Describe the meaning and objective of data independence.

(2 marks)

ANSWER POINTER

Data independence means that upper levels of the architecture are unaffected by changes to lower levels. Mainly, it aims to separate a user's view of the database from the way the database is physically represented.

(ii) Describe each of the three levels of the ANSI-SPARC architecture.

(6 marks)

ANSWER POINTER

- External level: users' view of the database. Describes the part of the database that is relevant to each user.
- Conceptual level: The community view of the database. Describes what data is stored in the database and the relationship among the data.
- Internal level: the physical representation of the database on the machine. Describes how the data is stored in the database.

Examiner's Comment

In general, candidates answered this part correctly. However, there was a significant number who confused the various levels to the three-tier architecture.

b) Database Management Systems provide the following services:

- Concurrent Control
- Recovery
- Authentication
- Integrity

Briefly describe each of the above services and describe how they can be achieved.

(8 marks)

ANSWER POINTER

- Concurrency control: this is a service that addresses conflicts that can occur when multiple users try to simultaneously access or alter data. Can be achieved through, for example, the use of locks, timestamping…
- Recovery: is the process of restoring the database and the data to a consistent state. It can be achieved through, for example, the use of logs, backups…
- Authentication: the process of checking that a user (person, application or process) is who he/it claims to be. Usually implemented through usernames and passwords.
- Integrity: safeguarding the accuracy and completeness of assets. Can be achieved through, for example, entity, referential and domain constraints.

Examiner’s Comment

Again, most candidates did a good job of describing these DBMS features. However, many either came up with a good description but failed to identify the corresponding mechanisms to implement those features or the other way around. In addition, many confused authentication with authorisation and started writing about privileges..

c) The three-tier architecture is commonly used to implement a database driven web application.

(i) Draw a diagram to illustrate this architecture.

(3 marks)

ANSWER POINTER
Diagram required showing the three tiers: client, web/application server, and database server.

(ii) Describe the role of each tier.

(3 marks)

ANSWER POINTER
The client: web browser (presentation layer) handles user input and renders the results of user queries in the form of HTML

Web/application server: implements business logic of the application; serves as the middle tier between client and database (retrieves results, validates input)

The database server: holds the DBMS and executes user queries.

(iii) Discuss three advantages of this architecture.

(3 marks)

ANSWER POINTER

Examples of advantages – only two required:

- Thin client
- Centralised maintenance (web or database server)
- Easy to replace one tier in isolation of other tiers

Examiner’s Comment

Most candidates had a good attempt at this part.