BCS THE CHARTERED INSTITUTE FOR IT

BCS HIGHER EDUCATION QUALIFICATIONS
BCS Level 5 Diploma in IT

DATABASE SYSTEMS

Tuesday 27th March 2018 - Morning
Answer 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

GENERAL COMMENTS

This was a very popular question with most candidates making an attempt. However, the number of attempts achieving pass level was a little lower than expected.

a) Explain the MAIN objectives of the ANSI-SPARC architecture for a DBMS. Discuss briefly the challenges of achieving these objectives in practice.

(10 marks)

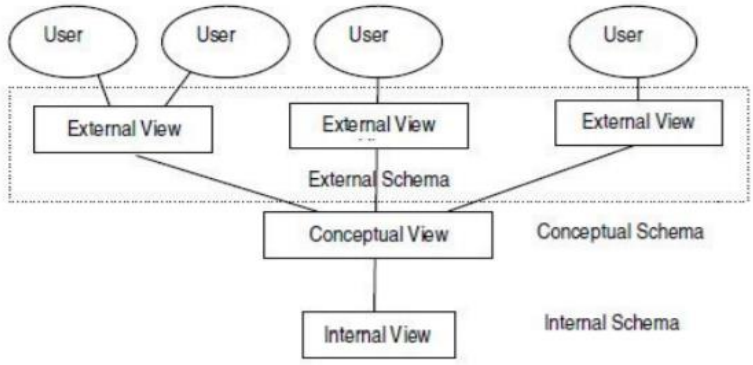
b) Nowadays, many organisations have chosen to have their data resources and services managed remotely by hosting databases on the 'Cloud'.

Describe the concept of the 'Cloud' and explain how it differs from a client-server database architecture. Comment on the advantages of running an organisation's database resources and services on the 'Cloud'.

(15 marks)

ANSWER POINTERS

Part a)
The objective of the ANSI-SPARC three-level architecture is to separate the user's view into three levels. A diagram is useful thus:



In the ANSI/SPARC architecture, a database can be seen at three levels, known respectively as physical, logical, and view. For each level, there is a schema: internal, conceptual, and external. The three levels are used to support Logical and Physical Data Independence

The logical level is concerned with the logical model and models information about how data is viewed by a user. For example, a table (relation) stored in the database and all its constraints, applied on that relation is a logical model of a Relational database. Alternative logical models exist.

Logical data independence eliminates the need to know how actual data is stored on the disk. If we make some changes on table format, it should not change the data residing on the disk.

Physical data independence has the power to change the physical data without impacting the schema or logical data. The physical data layer is intended to be hidden from view from users and independent of the logical layer. This view carries specific features concerned with index placement, page sizes and so on of the storage system.

Physical data independence allows independent user views: Each user should be able to access the same data, but have a different customized view of the data. These should be independent and changes to one view should not affect others.

In addition, it hides the physical storage details from users: Users should not have to deal with physical database storage details.
The internal structure of the database should be unaffected by changes to the physical aspects of the storage: For example, a changeover to a new disk.
.
The ANSI/SPARC report was intended as a basis for interoperable computer systems. All database vendors adopted the three-schema terminology, but they implemented it in incompatible ways. Various groups attempted to define standards for the conceptual schema and its mappings to databases and programming languages. Unfortunately, none of the vendors had a strong incentive to make their formats compatible with their competitors'.

Part b)

The concept of the Cloud is a type of computing that is delivered by external computing services – servers, storage, databases, networking, software, analytics and more – over the Internet ("the cloud"). Companies offering these computing services are called cloud providers and typically charge for cloud computing services based on usage, similar to how billing is carried out for gas or electricity at home. The service can range from software products only, to full infrastructure of a distributed computing db system.
A client server db system is a traditional way in which an organisation would host its own database and service the needs of users by providing a client (desktop computer) to access a remote server. Here connections can be achieved via WWW (http) or via a local network such as an intranet. Organisations are generally trending towards hosted services for the reasons outlined below unless there is a specific type of application that needs to be controlled/accessed 'in-house'
(5 marks)

Advantages of hosting DB services on the cloud
1. Cost
Cloud computing eliminates the capital expense of buying hardware and software, and setting up and running on-site data centres – the racks of servers, the round-the-clock electricity for power and cooling, the IT experts for managing the infrastructure. It adds up quickly.

2. Speed
Most cloud computing services are provided as self service and on demand, so even vast amounts of computing resources can be provisioned in minutes, typically with just a few mouse clicks, giving businesses a lot of flexibility and taking the pressure off capacity planning.

3. Scaling up
The benefits of cloud computing services include the ability to scale elastically. In cloud speak, that means delivering the right amount of IT resources – for example, more or less computing power, storage, bandwidth – exactly when it's needed, and from the right geographic location.

4. Productivity
On-site data centres typically require a lot of "racking and stacking" – hardware setup, software patching and other time-consuming IT management chores. Cloud computing removes the need for many of these tasks, so IT teams can spend time on achieving more important business goals.

5. Performance
The biggest cloud computing services run on a worldwide network of secure data centres, which are regularly upgraded to the latest generation of fast and efficient computing hardware. This offers several benefits over a single corporate data centre, including reduced network latency for applications and greater economies of scale.

6. High db integrity
Cloud computing makes data backup, disaster recovery and business continuity easier and less expensive, because data can be mirrored at multiple redundant sites on the cloud provider's network.

EXAMINERS' COMMENTS

Part a)
Many candidates produced long verbose descriptions of ANSI-SPARC mainly from bookwork and failing to answer what the question was asking. There was too little discussion of the objectives and challenges of realising the ANSI-SPARC architecture.
A diagram and a brief description were generally produced but this attracted only limited credit. A satisfactory discussion of the challenges of realising this architecture was credit worthy but this was very poorly answered and again marks were lost as a result.

Part b)
This part was fairly well answered (much better than the first part) and was broken down to the three subparts as shown above. Most of the marks were awarded for the last subpart where insightful comments about the advantages were required. The evidence shows that listing facts without comment of the advantages of cloud hosting was fairly common and this contributed to lower than expected marks.

A2

The integrity of database transactions must be maintained in a highly concurrent multi-user online transaction processing environment.

Given the above context, describe the techniques a DBMS uses to maintain database integrity.

For guidance, your answer must address, with the aid of relevant examples, the following five topics:

a) What function is performed by a **TYPICAL** database transaction compared to a program that reads/writes to a traditional file-based system?

b) How database integrity is affected by concurrent transactions.

c) Why database transactions have to be isolated to preserve database integrity.

d) What happens when transactions have to be aborted?

e) How the DBMS recovers transactions that are lost following system failure or crashes.

(25 marks)

ANSWER POINTERS

A database transaction performs the function of reading or writing from/to a database tables and consists of a series of data manipulation statements usually written in SQL. Database transactions differ from file-based systems in that they must either fully complete or fully fail, leaving the database in a consistent state. A traditional program that reads and writes to a file requires a different mechanism to check for integrity (usually at the application level using exception handlers – and the routine might then only handle basic operations on the state of a file, for example open; not found

Each statement can change the state of the database and so must adhere to strict rules to prevent inconsistency and loss of data integrity.
The strictness is enforced by ensuring each transaction adheres to certain rules known as the ACID rules.

In a concurrent environment db access has to be managed to prevent inconsistencies. For example, a row in tableA might be updated by transaction A and at the same time read by transaction B. If the data in tableA is not locked during the update, transaction B may read inconsistent data. Isolated transactions that interleave with each other are the focus of the I in the ACID properties of transactions.

When a transaction is aborted (due to failure) code is written to accommodate rollback of transactions that occurred in that transaction. The effect is to return the database to a consistent state that existed before the error that was detected occurred.

Reprocessing of transactions that succeeded previously may need to be performed if there was some catastrophic failure in the DBMS with permanent loss of data. The only way to ensure full recovery will be to access the transaction log (a record saved on backup of previous successful transactions) and replay transactions from a savepoint a recorded and stable state that the database was in prior to failure.

EXAMINER'S COMMENTS

About half of the candidates attempted this question. The average mark was lower than expected with less than half of the attempts achieving pass level marks. The evidence shows that a number of candidates failed to follow the guidance provided, which was intended to direct candidates to cover all the important points. Even when using the guidance many candidates produced answers that were not always applicable or went into unnecessary detail. The most noticeable weakness which should be of concern was a demonstration of a lack of understanding of the profound nature of database transactions compared with traditional file based transactions that simply read/write to a file.

A3

GENERAL COMMENT

This was a very popular question with almost all candidates making an attempt. However, less than half of the answers achieved pass level.

a) Each member of staff in a company is allocated a single computer. Information about computers belonging to staff members is stored in a database consisting of the following table:

HasComputer (StaffNbr, ComputerNbr, Manufacturer, Cost)

Company policy subsequently changes and some staff are allocated more than one computer (Each computer is still assigned to a single member of staff).

(i) Explain why the above table design is no longer appropriate.

(1 mark)

(ii) Suggest a new database design to handle the new situation.

(4 marks)

ANSWER POINTERS

(i) The original design is no longer appropriate because (accept either)
- It requires multiple entries with the same primary key, or
- It introduces repeating groups into the table.

(ii) The solution is to remove the repeating group to a new table, thus.
HasComputer(StaffNbr, ComputerNbr) – accept answer with composite key
ComputerDetails(ComputerNbr, Manufacturer, Cost)

EXAMINERS' COMMENTS

Most candidates managed to split the original table into two tables, but many failed to identify that `StaffNbr` could not be a primary key.

b) Each member of staff in a company works on a single project. Several staff work on each project, and all staff working on the same project share the same office. The following table stores information about staff, projects and offices.

Works (StaffNbr, ProjectNbr, ProjectDescription, OfficeNumber)

(i) Explain how an update anomaly could arise in the above table.

(2 marks)

(ii) Identify the underlying fault in the above design and show how the table can be transformed to remove this flaw.

(6 marks)

(iii) Suppose the situation is as described above, except that now each project has just ONE member of staff working on it. Is the original table design still faulty? Explain your answer.

(2 marks)

ANSWER POINTERS

(i) If we wish to change an attribute for a project e.g. its Office Number, then this has to be changed separately for each staff member working on that project. Mistakes or omissions could then lead to inconsistent data.

(ii) The underlying fault is that Project Description and Office Number are transitively dependent on StaffNbr via ProjectNbr
The table can be transformed by moving the Project information from the Works table into a new table:

Works(StaffNbr, ProjectNbr)
Project(ProjectNbr, ProjectDescription, OfficeNumber)

(iii) If each project has just one member of staff working on it, then the original table design is not faulty.
This is because now each project can only appear once in the table, so there is no data duplication, and hence no opportunity for anomalies.

EXAMINERS' COMMENTS:

Although many candidates managed to normalise the original table, most did not recognise the existence of transitive dependencies.

(c) The following table keeps records of medical consultations conducted in a medical practice. Each consultation takes place in a room and is conducted by a doctor on a patient. A patient cannot have two consultations on the same day.

patientNbr	consultationDate	consultationTime	doctorNbr	roomNbr
P01	13-Sep-17	08:30	D01	R01
P01	18-Sep-17	15:00	D02	R02
P02	13-Sep-17	10:00	D01	R01
P03	13-Sep-17	10:00	D02	R02
P04	26-Sep-17	08:30	D01	R02

(i) Explain what is meant by a “functional dependency” between attributes in a table.

(2 marks)

(ii) Identify two functional dependencies from the above table.

(6 marks)

(iii) Suggest a suitable primary key for the above table.

(2 marks)

ANSWER POINTERS

(i) Given attributes A and B of a relation (A and B may each consist of one or more attributes), a functional dependency exists between A and B, denoted A → B, when A uniquely determines B (each value of A will always correspond to the same value of B).

(ii) (patientNbr, consultationDate) → doctorNbr
(doctorNbr, consultationDate) → roomNbr

(iii) (patientNbr, consultationDate)

EXAMINERS' COMMENTS

There is evidence that candidates struggled to understand the concept of functional dependency. Only a handful managed to provide correct answers.

Section B
Answer Section B questions in Answer Book B

B4

GENERAL COMMENTS

This was a very popular question. The vast majority of candidates made an attempt.

A **database designer** is bidding for a new contract with a prestigious blue-chip organisation. Part of the selection process is a technical interview. Answer the following questions from the interview panel.

(a) Explain, with examples, how a database table may be logically connected to another table, including the associated rules demanded by *referential integrity* to support such connectivity and any subsequent changes in either table.

(10 Marks)

ANSWER POINTERS

Marked holistically with bonus marks for concrete examples, but a strong response will address the term 'foreign key', atomic and composite foreign keys, the concepts of 'referenced' and 'referencing' tables, the need for the foreign key to map onto a comparable (in terms of domain) 'candidate' key - normally the primary key, but it does not have to be. The fact that a given foreign key value must either be NULL (point 'nowhere') or have a value that exactly matches a value in the 'referenced' table. It cannot 'point' at a non-existing value. The four options available to the 'referencing' table when a 'referenced' table value/row is deleted/updated should also be covered:

- RESTRICT (disallow modification)
- NULLIFY (allow modification but insert a NULL entry)
- DEFAULT (allow modification but insert a dummy value)
- DELETE CASCADE (allow modification and delete any dependent rows in other table)

EXAMINER'S COMMENTS

This sub-question focused on referential integrity and foreign keys (FK) and was generally answered very well by most candidates. All attempts covered the issues of linking to another table via a primary key (PK) – although only a very small number distinguished between a primary key and a candidate key. All Included discussions of the need to match the FK to the PK and most (but not all) went on to talk about how the referential integrity mechanism allows a FK to be duplicated and to hold NULL entries. Only a handful mentioned the `ON DELETE` and `ON UPDATE` issues such as CASCADE, restrict, insertion of NULL etc. Most candidates supplied quality diagrams with clear annotation. This was a good topic for most.

(b) Explain, with examples, how the rows in a given database table may be uniquely identified, including the associated rules demanded by *entity integrity* and what guidelines exist for the selection of such a row-identification mechanism.

(10 Marks)

ANSWER POINTERS

A strong response will address the two entity integrity rules for any candidate key – no duplicates values, no nulls entries. Each value in a candidate key must be a real, unique value. A given table may possess many candidate keys but the one chosen becomes the primary key, the others becoming alternate keys. Primary keys should be as minimal as possible – so if one candidate key has three attributes and another one attribute, the latter should be adopted as the primary key. This leads into the concepts of atomic and composite primary/candidate keys and why, in reality, primary keys are 'man-made or 'surrogate' attributes like serial number or student ID, rather than 'natural' attributes such as name, address, date of birth etc. This use of minimal, surrogate keys enables auto-generation of such keys found in real-world databases – like ISBN, NI Numbers, NHS numbers etc.

EXAMINER'S COMMENTS

This sub-question addressed entity integrity and primary keys. The obvious points about disallowing duplicated and NULL entries were documented by all candidates. The stronger responses did move on to discuss atomic and composite PKs but again, few differentiated between candidate and primary keys. On the whole, a well answered question.

(c) Show how appropriate relational algebra operations would be used to extract
(i) specific tuples.
(ii) specific attributes.
(iii) specific attributes of specific tuples from a populated database table. **(5 Marks)**

ANSWER POINTERS

A good response will explain the role of SELECTION as a row-oriented 'horizontal' filter that targets tuples and PROJECTION as a column-oriented 'vertical' filter that targets attributes. They can of course be combined. The better candidates will stress that both RA concepts are actually implemented in SQL using the `SELECT...FROM...WHERE` queries:

- **PROJECTION (of all rows from three named columns)**:
```
SELECT ID, Name, Email FROM Student;
```
- **SELECTION (of all columns for those rows in the specified subjects):**
```
WHERE Subject IN ('Computing', 'Computer Science', 'IT');
```
- **COMBINED**:
```
SELECT ID, Name, Email FROM Student
WHERE Subject IN ('Computing', 'Computer Science', 'IT');
```

EXAMINER'S COMMENTS

This final section covered Relational Algebra - specifically the selection and projection operators. This was by far the most difficult topic for most candidates. Many candidates did not mention it while others discussed general concepts such as attributes and tuples. Relatively few candidates focussed on the specific Relational Algebra operations of selection and projection. Those that did provided good answers supported by clear examples.

B5

GENERAL COMMENTS

This was not a popular question, attempted by a (sizable) minority of candidates.

A **SQL developer** is bidding for a new contract with a prestigious blue-chip organisation. Part of the selection process is a technical interview. Answer the following questions from the interview panel:

a) For each of the following terms, explain what the term stands for, the essence of the functions it provides and a set of example SQL statements (at least TWO for each) that implement these functions:

(i) DDL. **(5 Marks)**
(ii) DML. **(5 Marks)**
(iii) DCL. **(5 Marks)**

ANSWER POINTERS

(i) DDL – Data Definition Language – the part of SQL responsible for all things 'structural' such as `CREATE/ALTER/DROP` with tables, views, indexes etc. Builds the underpinning database components.

(ii) DML – Data Manipulation Language - the part of SQL responsible for adding, modifying and removing content (rows) from tables and views. Uses the `INSERT/UPDATE/DELETE` statements.

(iii) DCL – Data Control Language - the part of SQL responsible for all things related to 'restrictions' – normally meaning the issuing and removal of privileges to perform certain database actions. This is the security aspect of DCL and uses `GRANT` & `REVOKE`. It could be argued that data integrity is on the border between DDL and DCL with clauses like `NOT NULL`, `UNIQUE` and `CHECK` being both structural components (DDL) and restriction components (DCL).

EXAMINERS' COMMENTS

This section concerned the concepts of DDL, DML and DCL aspects of the SQL database language. Almost every candidate correctly identified the full name of each acronym and then went on to explain the key concepts behind it, usually supported with at least one clear SQL example. This was a good topic for the majority of candidates who attempted it.

b) Using your own specific examples, illustrate the following SQL querying techniques:

(i) Row-level filtering. **(2 marks)**

(ii) Aggregate functions. **(2 marks)**

(iii) Grouping and group-level filtering. **(2 marks)**

(iv) Joining. **(2 marks)**

(v) Sub-Queries. **(2 marks)**

ANSWER POINTERS

(i) Example of use of `WHERE` clause.
(ii) Examples of `MIN/MAX/COUNT/AVG/SUM`.
(iii) Examples of `GROUP BY` and `HAVING` clauses.

(iv) Comparisons of `RECURSIVE/NATURAL/INNER/OUTER (FULL/LEFT/RIGHT)`.

(v) Examples of nested queries and `IN/NOT IN/EXISTS` operators.

EXAMINERS' COMMENTS

This section examined specific SQL querying techniques (filtering, aggregation, grouping, joining and sub-queries). It was less well answered than part (a), with many of the five areas being omitted. Most candidates submitted a single piece of SQL, but it was often not syntactically correct. Few supported the SQL code with any type of supporting explanations. That said, it was clear to see that across the five topics, most candidates understood the key ideas behind SQL techniques and were rewarded as such.

B6

GENERAL COMMENT

Just under a half of the students attempted this question. The pass rate was only 27%, with an average mark of 7 and a standard deviation of 4.5.

(a) Database management systems have to make provision to recover from situations such as system crashes.

(i) Explain what is meant by a *log* file, its typical content, and why it is required for recovery.

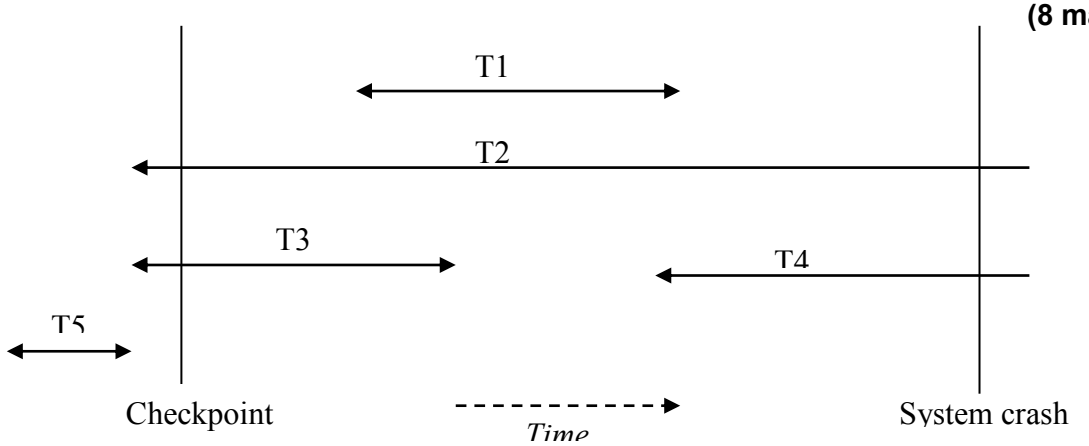
(3 marks)

(ii) Explain what is meant by the *deferred update recovery protocol*.

(2 marks)

(iii) Consider the following series of transactions, which you may assume took place using the *deferred update recovery protocol*. The arrowheads represent the start and end points of the transactions. For each transaction, explain how the recovery process deals with that transaction.

(8 marks)



ANSWER POINTERS

(i) A system failure may result in a transaction's results being lost. To be able to restore them, there has to be a system log which records the changes. In the event of a system failure, the log can be used by the DBMS to redo the effects of the transaction. A log will include transaction records (id, start time, read/write operations, before/after image …) and checkpoint records.

(ii) Under the deferred update recovery protocol, updates to the log and the database are only written to disk once the transaction commits. The log is written first.

(iii) The checkpoint causes T5 to be written to disk, so it isn't affected by the crash, and doesn't require to be recovered
T2 and T4 have not committed at the time of the crash, so under the deferred update protocol, they will have written nothing to disk. Hence the recovery process can ignore them. They can be re-run later if required.
T1 and T3 have committed at the time of the crash, but the full results may not yet have been written to disk. Committed results must not be rolled-back, so we are required to re-do T1 and T3 from the log. The log includes the time order of writes, and the final value in each case, so provides the necessary information to update the variables affected.

EXAMINERS' COMMENT

Almost all candidates failed to provide correct answers to this question.

(b) The following table gives information about staff in a company:

StaffID	**StaffName**	**ProjectNbr**	**PhoneNbr**	**Salary**
S1	Fred	P3	1234	20000
S2	Bill	P3	1235	25000
S3	George	P5	1267	19000
S4	Jim	P5	1294	19500

It has been decided to develop a number of views based on this table for the convenience of the users. Give an example of a horizontal view of this table, and an example of a vertical view, and explain why each might be useful. (In each case, just list the contents of the view; you are not required to write any SQL, for example CREATE VIEW statements).

(4 marks)

ANSWER POINTERS

The following is a horizontal view showing all details of staff working on project P3. Such a view would be useful for e.g. the project manager of this project

StaffID	**StaffName**	**ProjectNbr**	**PhoneNbr**	**Salary**
S1	Fred	P3	1234	20000
S2	Bill	P3	1235	25000

(1 mark for any horizontal view + 1 mark for any sensible purpose)

The following is a vertical view, showing all staff, but omitting the Salary attribute.
This would allow members of the organisation to see non-confidential information, while hiding the Salary data, which would normally be confidential.

StaffID	**StaffName**	**ProjectNbr**	**PhoneNbr**
S1	Fred	P3	1234
S2	Bill	P3	1235
S3	George	P5	1267
S4	Jim	P5	1294

(1 mark for any vertical view + 1 mark for any sensible purpose)

EXAMINERS' COMMENTS

There were some good attempts at this question, despite some not able to distinguish horizontal from vertical.

(c) Relational databases are very effective in situations for which they are appropriate. In other situations, simpler file-based solutions may be sufficient. Suppose you are required to implement a system for storing information about a library's books, borrowers, and loans. Give FOUR reasons why a database system is superior to a file-based system for this task. Illustrate the answer with suitable examples.

(8 marks)

ANSWER POINTERS

1 mark per reason + 1 mark per illustration (up to a maximum of 8 marks). Possible reasons and illustrations include:

- Less duplication of data leading to reduced risk of inconsistent data. E.g. if a book is loaned, then it must be in the Book table.
- Data can be shared between users. E.g. managers, librarians, borrowers all use the same DB, accessing whichever bits are of interest to them.
- Easy to query multiple tables. E.g. addresses of borrowers reading books on a certain topic.
- Data can be checked for validity more easily. E.g. ISBN must be valid; return date must be 1 month later than check-out date.
- Allows different users to access and view different parts of the DB. E.g. librarians need different views from borrowers.

EXAMINERS' COMMENTS

Almost all candidates recognise the fact that databases offer a more efficient way of managing and querying records. However, there is evidence that many candidates mistake the file-based approach with physical files and cabinets instead of a computerised file system.

END OF EXAM