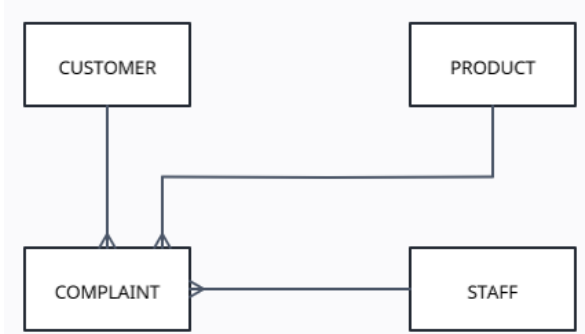


4(a)	1 mark per bullet point, max 2 marks - The relationship between SHIP and CONTAINER is one-to-many (1:M) - The primary key <u>ShipID</u> in the SHIP table is linked to the foreign key ShipID in the CONTAINER table
4(b)	1 mark per bullet point, max 2 marks - ALTER TABLE statement - Including the additional field with a suitable field name and suitable field type Example answer <pre>ALTER TABLE CONTAINER ADD InspectionDate DATE;</pre>
4(c)	1 mark per bullet point, max 4 marks - SELECT <u>COUNT</u> statement - Using the correct tables - Joining the tables - The condition for the name Example Answer 1: <pre>SELECT COUNT(ContainerID) FROM CONTAINER, SHIP WHERE CONTAINER.ShipID = SHIP.ShipID AND ShipName = "Caledonia";</pre> Example Answer 2: <pre>SELECT COUNT(ContainerID) FROM CONTAINER INNER JOIN SHIP ON CONTAINER.ShipID = SHIP.ShipID WHERE ShipName = "Caledonia";</pre>
4(d)	1 mark per bullet point, max 2 marks - To allow the user to create / modify / delete tables // maintain the database - To allow the user to set up / modify relationships - To allow the user to create a form for data input - To allow a user to add tools to a form // for example, drop-down boxes / buttons etc. - To allow a user to design a report (to show the output in an organised manner) - To allow a user to add a menu to enable a choice of different actions - To allow a user to inspect the database contents and metadata ... by creating and/or running different SQL queries

5(a)	1 mark for each correct relationship, max 3 marks 
5(b)	1 mark per bullet point, max 4 marks - Create table with opening and closing brackets and commas separating the attributes - Appropriate data types for StaffFirstName, StaffLastName and Department - Appropriate data types for StaffID and RemoteWorker - Primary key correctly defined Example Answer 1: <pre>CREATE TABLE STAFF(StaffID INTEGER, StaffFirstName VARCHAR, StaffLastName VARCHAR, Department CHAR, RemoteWorker BOOLEAN, PRIMARY KEY (StaffID));</pre> Example Answer 2: <pre>CREATE TABLE STAFF(StaffID INTEGER NOT NULL PRIMARY KEY, StaffFirstName VARCHAR, StaffLastName VARCHAR, Department CHAR, RemoteWorker BOOLEAN);</pre>

5(c)	1 mark per bullet point, max 5 marks • SELECT and the correct attributes • FROM and the correct tables • Tables joined correctly • Correct condition for the rating • Correct ORDER BY clause Example Answer 1 SELECT PRODUCT.ProductID, ProductName, ComplaintDetails FROM PRODUCT, COMPLAINT WHERE PRODUCT.ProductID = COMPLAINT.ProductID AND Rating <=5 ORDER BY Rating DESC; Example Answer 2 SELECT PRODUCT.ProductID, ProductName, ComplaintDetails FROM PRODUCT INNER JOIN COMPLAINT ON PRODUCT.ProductID = COMPLAINT.ProductID WHERE Rating <=5 ORDER BY Rating DESC;
5(d)	One mark for identification of method one mark for corresponding description max 4 marks • Authentication methods / passwords / biometrics / 2-factor authentication can be implemented • ... which prevents unauthorised access to the customer's data • Access rights / privileges can be set • ... so that only those with correct permissions can read / edit customer's data • Regular backups can be scheduled • ... so that a second copy of the customer's data is available in case of loss/damage • The data can be encrypted • ... so that the customer's data cannot be understood by anyone who gains unauthorized access • Different views can be created • ... so that not everyone can see the customer's data

5(a)	1 mark for each relationship. Max 2 if any extra <pre> graph TD CUSTOMER --> CUSTOMER_CARD_DATA CUSTOMER --> ORDER ORDER --> ORDER_ITEM </pre>								
5(b)	1 mark for: CardNumber								
5(c)	1 mark per row to max 2 <table border="1"> <thead> <tr> <th>Table</th><th>Foreign key</th></tr> </thead> <tbody> <tr> <td>ORDER_ITEM</td><td>OrderID</td></tr> <tr> <td>ORDER</td><td>CustomerID</td></tr> <tr> <td>CUSTOMER_CARD_DATA</td><td>CustomerID</td></tr> </tbody> </table>	Table	Foreign key	ORDER_ITEM	OrderID	ORDER	CustomerID	CUSTOMER_CARD_DATA	CustomerID
Table	Foreign key								
ORDER_ITEM	OrderID								
ORDER	CustomerID								
CUSTOMER_CARD_DATA	CustomerID								
5(d)	1 mark each to max 3 e.g. • Each order would only be able to have one item • or the database would not be normalised • it would not be in 1NF • due to repeated groups of attributes • in the ORDER table								
5(e)	1 mark each • Selecting the customer ID, customer name and sum of TotalCost with appropriate identifier • FROM clause with suitable join of tables (ON or WHERE) • ORDER.Paid = FALSE condition (with correct key word) • GROUP BY condition e.g. SELECT CUSTOMER.CustomerID, CUSTOMER.Name, Sum(ORDER.TotalCost) AS TotalOwed FROM CUSTOMER INNER JOIN ORDER ON CUSTOMER.CustomerID = ORDER.CustomerID WHERE ORDER.Paid = FALSE GROUP BY CUSTOMER.CustomerID; Alternative JOIN Statement SELECT CUSTOMER.CustomerID, CUSTOMER.Name, Sum(ORDER.TotalCost) AS TotalOwed FROM CUSTOMER,ORDER WHERE CUSTOMER.CustomerID = ORDER.CustomerID AND ORDER.Paid = FALSE GROUP BY CUSTOMER.CustomerID;								

6(a)	1 mark for each correct relationship <pre> graph LR CUSTOMER --> ORDER SUPPLIER --> PRODUCT ORDER --> ORDER_PRODUCT PRODUCT --> ORDER_PRODUCT </pre>
6(b)	1 mark each • INSERT INTO PRODUCT • VALUES with opening and closing brackets • Inserting ProductID, ProductName and SupplierID correctly as strings including quotation marks into correct fields • Inserting quantity and cost correctly into correct fields Example 1: INSERT INTO PRODUCT VALUES ("002323", "Blue ball point 2 mm", 50, 5.00, "SFX223"); Example 2: INSERT INTO PRODUCT (ProductID, ProductName, QuantityInBox, Cost, SupplierID) VALUES ("002323", "Blue ball point 2 mm", 50, 5.00, "SFX223");

6(c)	1 mark each • Selection of customer name and counting any field from ORDER as an appropriate identifier • Joining tables ORDER and CUSTOMER • AND (or WHERE) clause: Collected = FALSE • Grouping by customer ID or customer name Example 1: SELECT CustomerName, COUNT(OrderID) AS NotCollected FROM ORDER, CUSTOMER WHERE ORDER.CustomerID = CUSTOMER.CustomerID AND Collected = FALSE GROUP BY CUSTOMER.CustomerID Example 2: SELECT CustomerName, COUNT(OrderID) AS NotCollected FROM ORDER INNER JOIN CUSTOMER ON ORDER.CustomerID = CUSTOMER.CustomerID WHERE Collected = FALSE GROUP BY CUSTOMER.CustomerID
6(d)(i)	1 mark each to max 3 e.g. • Relationships • Views • Data types • Validation rules
6(d)(ii)	1 mark each to max 3 e.g. • To create / modify / delete database objects • To create a form for data input • To add tools to a form • for example, drop-down boxes / buttons etc. • To design a report to show the output in an organised manner • To add a menu to enable users to choose different actions / run different queries

2(a)	1 mark for limitation and 2 marks for corresponding explanation e.g. Data redundancy // data duplication • Separate linked tables are used • Data items are stored once so reduces data duplication / data redundancy Data inconsistency / Poor data integrity • Data changed once in one place will automatically update elsewhere • Linked data cannot be entered differently in two tables • Referential integrity can be enforced The data structure used depends on the application • Changes to the data structure are managed by the DBMS • ... and queries are not dependent on the structure of the data • ... and changes to the data do not require programs to be re-written
2(b)(i)	1 mark for each correct relation <pre> graph LR RP[REPAIR_PART] --> P[PART] R[REPAIR] --> C[CUSTOMER] RP --- R </pre>
2(b)(ii)	1 mark for each bullet point • Create table REPAIR_PART with opening and closing brackets, all statements inside the brackets • PartID and RepairNumber as varchar (or equivalent) • Quantity as integer • At least 1 appropriate constraint • Dual primary key For example: <pre> CREATE TABLE REPAIR_PART (PartID VARCHAR(20) NOT NULL, RepairNumber VARCHAR(4) NOT NULL, Quantity INT NOT NULL, PRIMARY KEY (PartID, RepairNumber)); </pre>

2(b)(iii)	1 mark for each bullet point • Select SUM of AmountDue • From the correct table and one correct condition • Second correct condition <pre> SELECT SUM (AmountDue) FROM INVOICE WHERE SupplierID = "JK675" AND Paid = FALSE; </pre>								
2(c)	1 mark for each correct definition <table border="1"> <thead> <tr> <th>Database Term</th><th>Definition</th></tr> </thead> <tbody> <tr> <td>Referential Integrity</td><td>All duplicate entries of data between tables are consistent // all foreign keys are matched to an appropriate primary key</td></tr> <tr> <td>Candidate Key</td><td>A field that could be a primary key but is not // an attribute or smallest set of attributes in a table where no tuple has the same value</td></tr> <tr> <td>Tuple</td><td>A row / record in a table // one instance of an entity in a table</td></tr> </tbody> </table>	Database Term	Definition	Referential Integrity	All duplicate entries of data between tables are consistent // all foreign keys are matched to an appropriate primary key	Candidate Key	A field that could be a primary key but is not // an attribute or smallest set of attributes in a table where no tuple has the same value	Tuple	A row / record in a table // one instance of an entity in a table
Database Term	Definition								
Referential Integrity	All duplicate entries of data between tables are consistent // all foreign keys are matched to an appropriate primary key								
Candidate Key	A field that could be a primary key but is not // an attribute or smallest set of attributes in a table where no tuple has the same value								
Tuple	A row / record in a table // one instance of an entity in a table								

6(a)	1 mark each to max 3 e.g. - There is reduced data redundancy // less repeated data ... because each item of data is only stored once - Data consistency is maintained // Data integrity is improved ... changes in one table will automatically update in another ... linked data cannot be entered differently in two tables - Program-data independence is ensured ... changes to the data do not require programs to be re-written // queries are not dependent on the structure of the data - Complex queries are easier to run - Different views can be provided so users can only see specific aspects of the database - Multiple concurrent access is possible ... through record locking
6(b)(i)	1 mark each to max 6 - CUSTOMER to JOB is 1 to many ... implemented by Primary Key in CUSTOMER is Foreign Key in JOB - EMPLOYEE to LOGIN_DATA is 1 to 1 ... implemented by Primary Key in EMPLOYEE is Foreign Key in LOGIN_DATA - JOB to JOB_EMPLOYEE is 1 to many ... implemented by Primary Key in JOB is Foreign Key in JOB_EMPLOYEE - EMPLOYEE to JOB_EMPLOYEE is 1 to many ... implemented by Primary Key in EMPLOYEE is Foreign Key in JOB_EMPLOYEE
6(b)(ii)	1 mark each - Select SUM of Amount - From the correct table and one correct condition - Remaining correct condition Example: <pre>SELECT SUM(Amount) FROM INVOICE WHERE Paid = "Y" AND DateSent >= #01/01/2023# AND DateSent <= #31/12/2023#</pre>

4(a)	1 mark each: - Foreign key: BatchID, table BATCH - Foreign key: CustomerID, table CUSTOMER
4(b)	1 mark each: - SELECT SUM(Quantity) - FROM SALE WHERE and one correct condition - AND with remainder correct conditions e.g. <pre>SELECT SUM(Quantity) FROM SALE WHERE CustomerID = "0034E" AND Date >= #01/01/2023# AND Date <= #31/12/2023#;</pre>
4(c)(i)	1 mark for each bullet point - Create table BATCH with opening and closing brackets, all statements within brackets - BatchID, Type and Flavour as varchar or equivalent ...with suitable constraint(s) - Size as decimal, SellingPrice as currency, EndDate as date (or equivalent) - Primary key identified as BatchID e.g. <pre>CREATE TABLE BATCH(BatchID VARCHAR(6) NOT NULL, Type VARCHAR (20) NOT NULL, Flavour VARCHAR (20) NOT NULL, Size FLOAT, SellingPrice CURRENCY, EndDate DATE, PRIMARY KEY (BatchID));</pre>

4(c)(ii)	1 mark for each bullet point each (max 4) e.g. - Ice cream table with an appropriate name ... containing type, flavour, size, selling price ... with suitable primary key ... foreign key identified in BATCH that links to the primary key in ICECREAM Example table definitions – not example answer: <pre>BATCH(BatchID, IceCreamID, EndDate) ICE_CREAM(IceCreamID, Type, Flavour, Size, SellingPrice)</pre>
4(d)	1 mark for each definition Entity: - A real-life object that is represented as a table Attribute: - An item of data about an entity
4(e)	1 mark for each bullet point each (max 3) - Referential integrity is enforced ... such as cascade update/delete // if the data is changed in one place it is updated in every other place ... and ensures each foreign key has a corresponding primary key

4(a)	1 mark for: 1-to-many
4(b)	1 mark each: - Creating table EXAM with opening and closing brackets - All fields with appropriate data types and commas at end of lines - ExamID as primary key Example: <pre>CREATE TABLE EXAM(ExamID varchar NOT NULL, Subject varchar, Level int, TotalMarks int, PRIMARY KEY (ExamID));</pre>
4(c)	1 mark each: - Altering table EXAM_QUESTION - Linking ExamID to ExamID in EXAM Example. <pre>ALTER TABLE EXAM_QUESTION ADD FOREIGN KEY (ExamID) REFERENCES EXAM(ExamID);</pre>
4(d)	1 mark each to max 5: - STUDENT table identified with suitable Primary Key - A linking table between STUDENT and EXAM with suitable Primary Key and appropriate name ... that includes the Primary Key of the STUDENT table as a Foreign Key to join with STUDENT ... and includes the Primary Key of the EXAM table as a Foreign Key to join with EXAM - A linking table between STUDENT and EXAM_QUESTION with suitable Primary Key and appropriate name ... that includes the Primary Key of Table 2 as a Foreign Key to join with Table 2 ... that stores the ExamQuestionID and the mark for that question