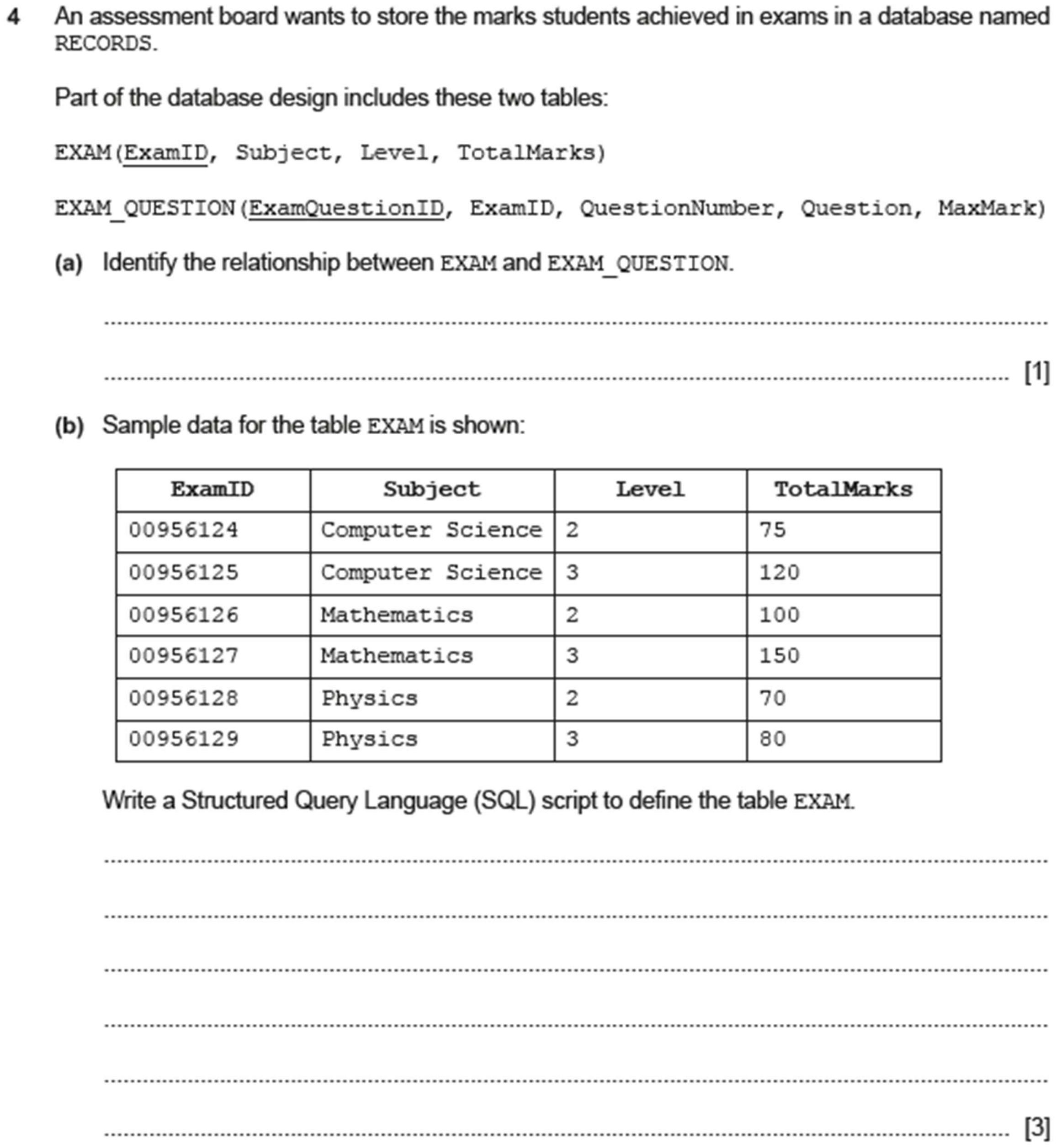
May/June 2024



4 An assessment board wants to store the marks students achieved in exams in a database named `RECORDS`.

Part of the database design includes these two tables:

`EXAM(ExamID, Subject, Level, TotalMarks)`

`EXAM_QUESTION(ExamQuestionID, ExamID, QuestionNumber, Question, MaxMark)`

(a) Identify the relationship between `EXAM` and `EXAM_QUESTION`.

....

.... [1]

(b) Sample data for the table `EXAM` is shown:

ExamID	Subject	Level	TotalMarks
00956124	Computer Science	2	75
00956125	Computer Science	3	120
00956126	Mathematics	2	100
00956127	Mathematics	3	150
00956128	Physics	2	70
00956129	Physics	3	80

Write a Structured Query Language (SQL) script to define the table `EXAM`.

....

....

....

....

....

.... [3]

(c) The table EXAM_QUESTION has been created but the foreign key has not been linked.

Write an SQL script to update EXAM_QUESTION **and** link the foreign key to EXAM.

....

....

....

.... [2]

(d) The database also needs to store data about the students, the exams the students have taken and the marks the students achieved in each question of each exam.

Describe the additional tables that will need to be included in the database **and** explain how all the tables in the database will be linked.

....

....

....

....

....

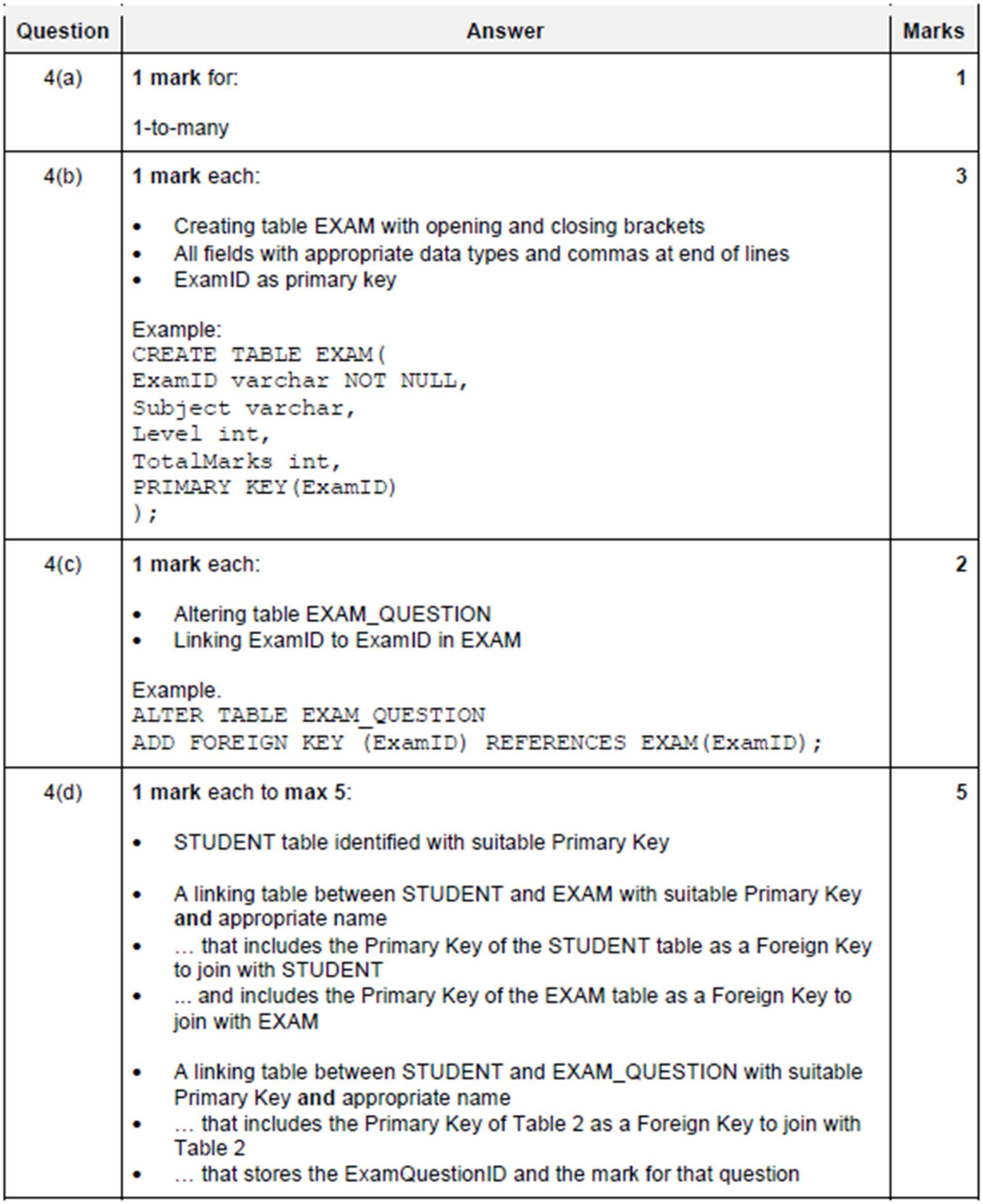
....

....

....

....

.... [5]



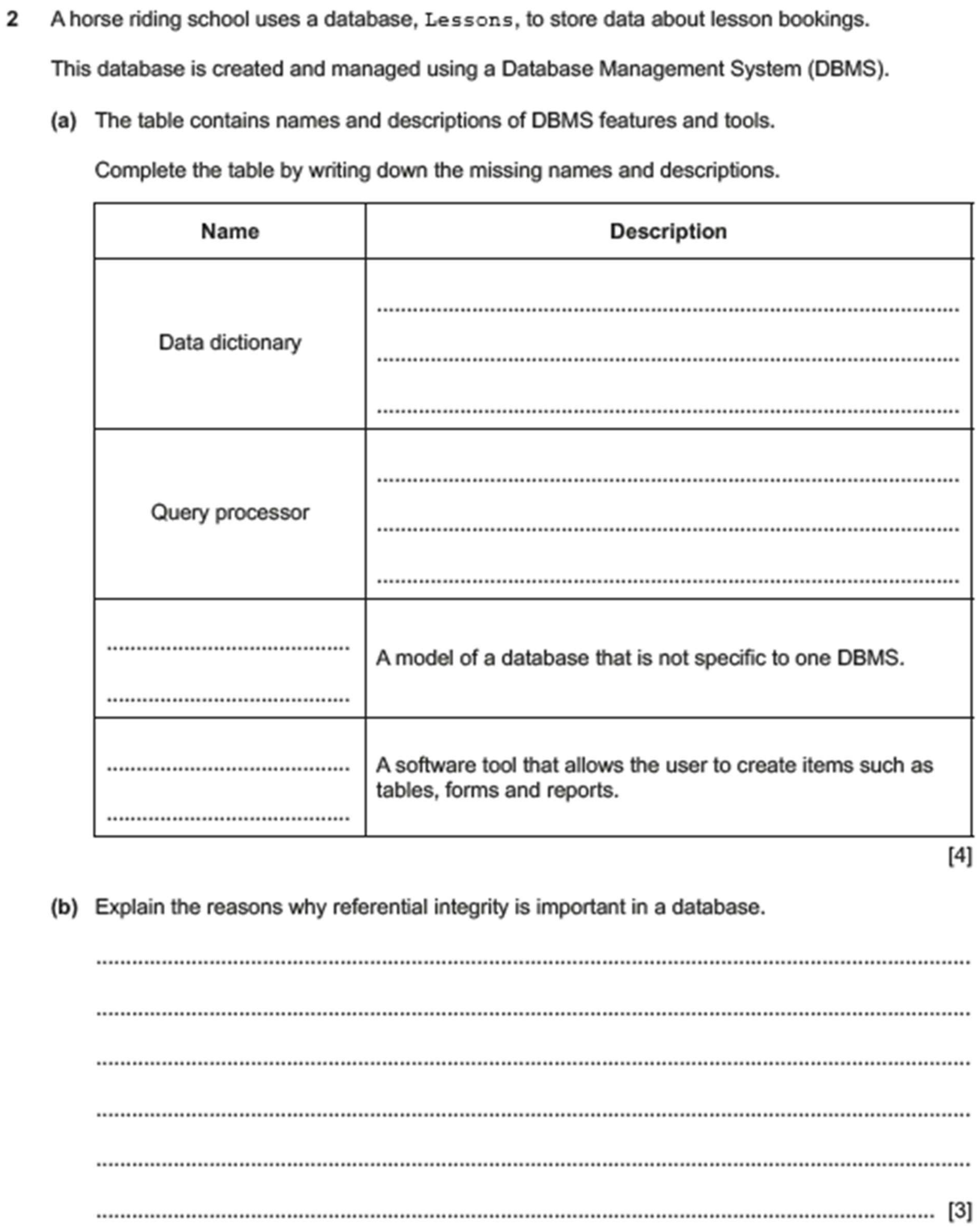
Question	Answer	Marks
4(a)	**1 mark** for: 1-to-many	1

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:
	`CREATE TABLE EXAM(`
	`ExamID varchar NOT NULL,`
	`Subject varchar,`
	`Level int,`
	`TotalMarks int,`
	`PRIMARY KEY(ExamID)`
	`);`
		3

4(c)	**1 mark** each:
	• Altering table EXAM_QUESTION
	• Linking ExamID to ExamID in EXAM
	Example.
	`ALTER TABLE EXAM_QUESTION`
	`ADD FOREIGN KEY (ExamID) REFERENCES EXAM(ExamID);`
		2

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	5

May/June 2023



2 A horse riding school uses a database, `Lessons`, to store data about lesson bookings.

This database is created and managed using a Database Management System (DBMS).

(a) The table contains names and descriptions of DBMS features and tools.

Complete the table by writing down the missing names and descriptions.

Name	Description
Data dictionary	....
Query processor	....
....	A model of a database that is not specific to one DBMS.
....	A software tool that allows the user to create items such as tables, forms and reports.

[4]

(b) Explain the reasons why referential integrity is important in a database.

....

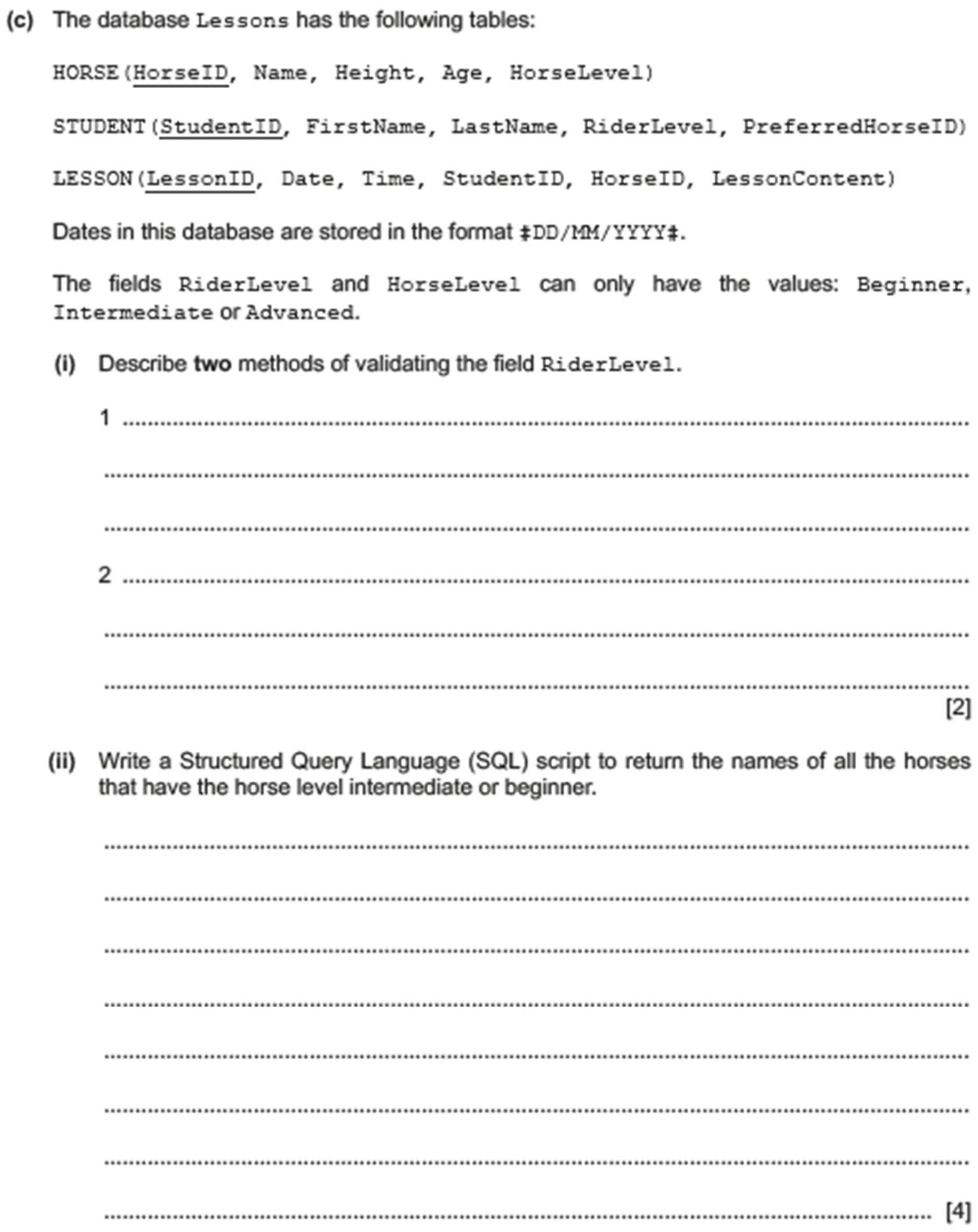
....

....

....

....

.... [3]



(c) The database `Lessons` has the following tables:

```
HORSE(HorseID, Name, Height, Age, HorseLevel)

STUDENT(StudentID, FirstName, LastName, RiderLevel, PreferredHorseID)

LESSON(LessonID, Date, Time, StudentID, HorseID, LessonContent)
```

Dates in this database are stored in the format `#DD/MM/YYYY#`.

The fields `RiderLevel` and `HorseLevel` can only have the values: `Beginner`, `Intermediate` or `Advanced`.

(i) Describe **two** methods of validating the field `RiderLevel`.

1 ....

....

....

2 ....

....

.... [2]

(ii) Write a Structured Query Language (SQL) script to return the names of all the horses that have the horse level intermediate or beginner.

....

....

....

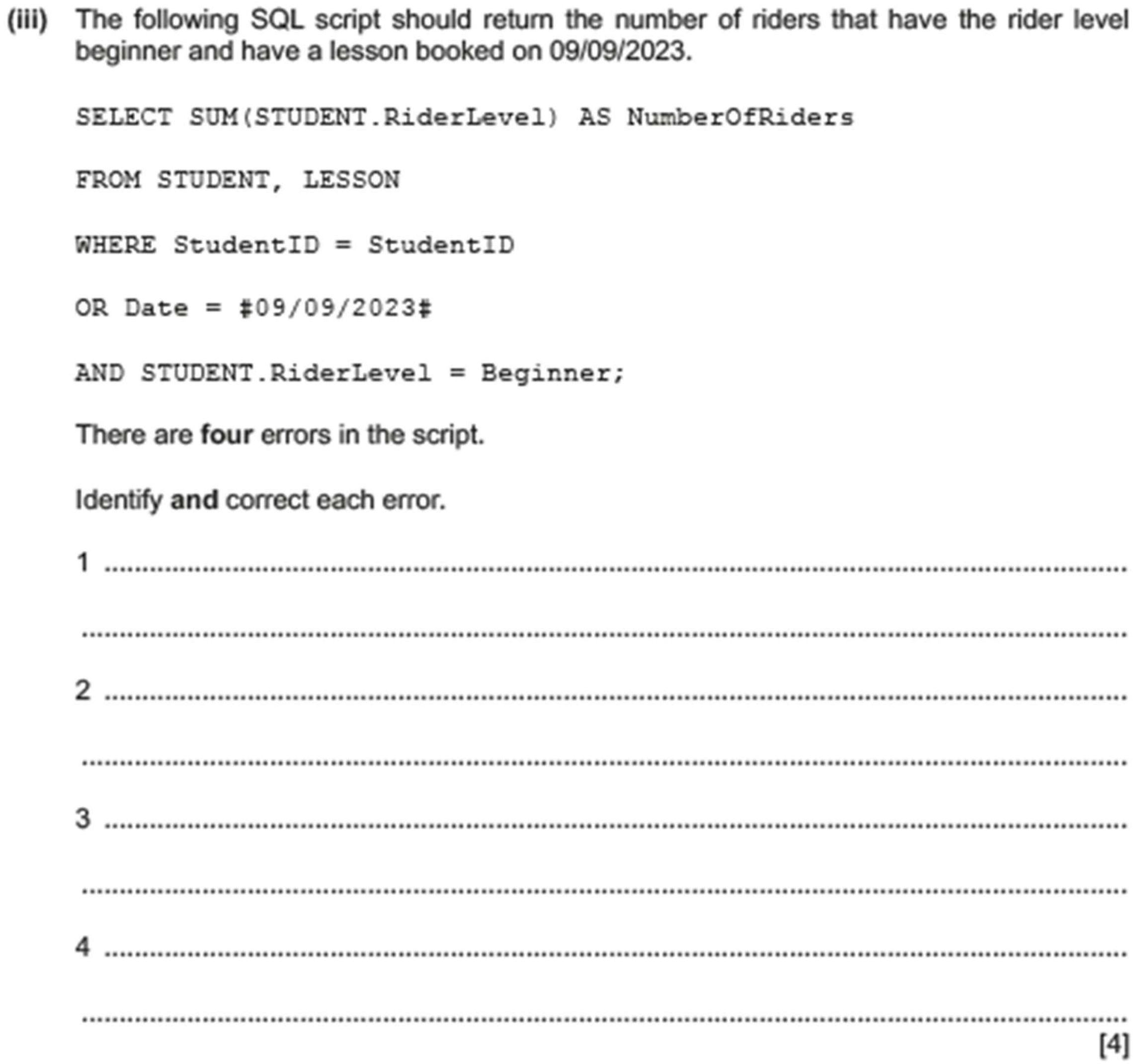
....

....

....

....

.... [4]



(iii) The following SQL script should return the number of riders that have the rider level beginner and have a lesson booked on 09/09/2023.

```
SELECT SUM(STUDENT.RiderLevel) AS NumberOfRiders

FROM STUDENT, LESSON

WHERE StudentID = StudentID

OR Date = #09/09/2023#

AND STUDENT.RiderLevel = Beginner;
```

There are **four** errors in the script.

Identify **and** correct each error.

1 ....

....

2 ....

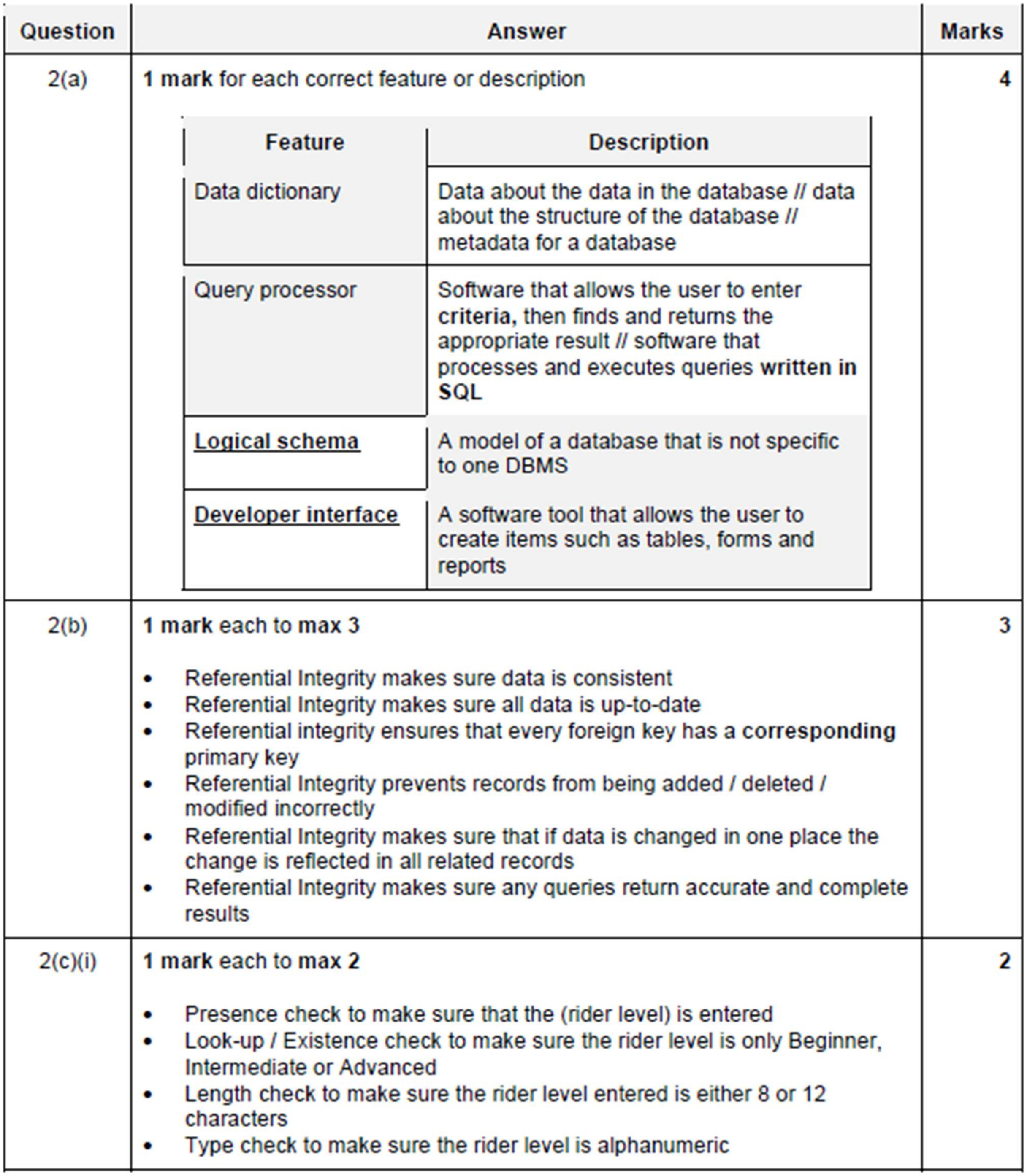
....

3 ....

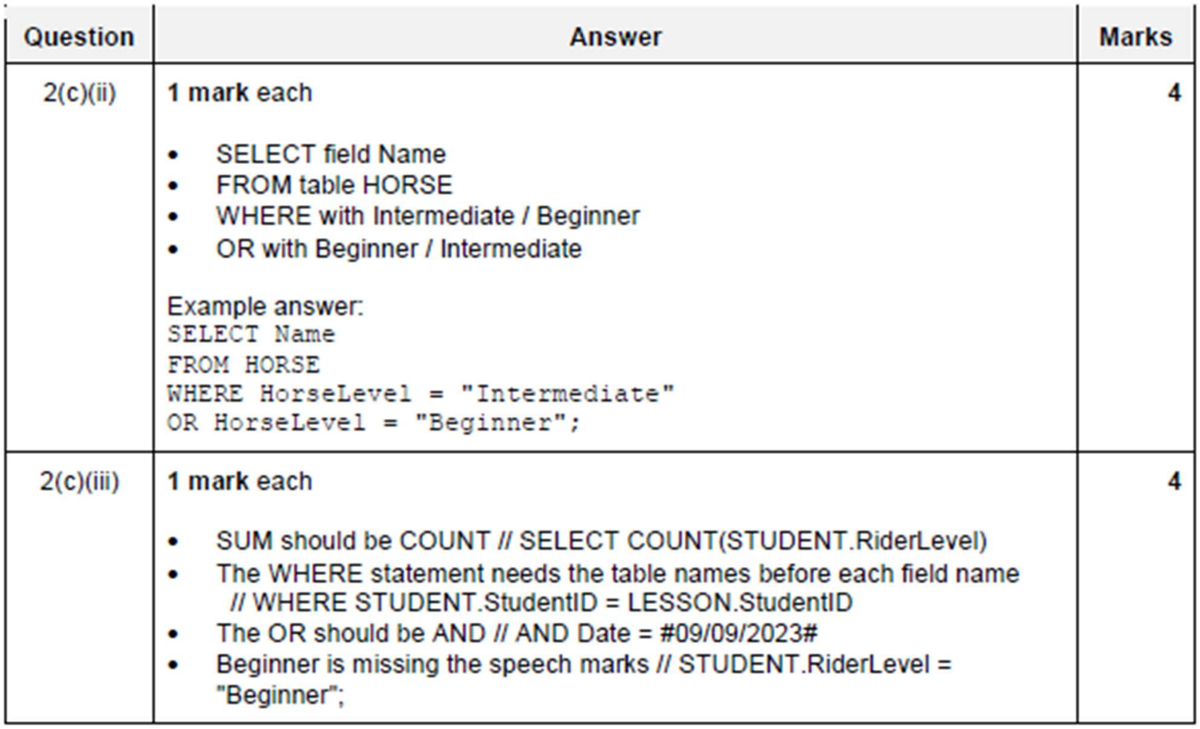
....

4 ....

.... **[4]**



<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>2(a)</td><td><b>1 mark</b> for each correct feature or description
<table>
<tr><th>Feature</th><th>Description</th></tr>
<tr><td>Data dictionary</td><td>Data about the data in the database // data about the structure of the database // metadata for a database</td></tr>
<tr><td>Query processor</td><td>Software that allows the user to enter <b>criteria,</b> then finds and returns the appropriate result // software that processes and executes queries <b>written in SQL</b></td></tr>
<tr><td><b><u>Logical schema</u></b></td><td>A model of a database that is not specific to one DBMS</td></tr>
<tr><td><b><u>Developer interface</u></b></td><td>A software tool that allows the user to create items such as tables, forms and reports</td></tr>
</table></td><td>4</td></tr>
<tr><td>2(b)</td><td><b>1 mark each to max 3</b>
• Referential Integrity makes sure data is consistent
• Referential Integrity makes sure all data is up-to-date
• Referential integrity ensures that every foreign key has a <b>corresponding</b> primary key
• Referential Integrity prevents records from being added / deleted / modified incorrectly
• Referential Integrity makes sure that if data is changed in one place the change is reflected in all related records
• Referential Integrity makes sure any queries return accurate and complete results</td><td>3</td></tr>
<tr><td>2(c)(i)</td><td><b>1 mark each to max 2</b>
• Presence check to make sure that the (rider level) is entered
• Look-up / Existence check to make sure the rider level is only Beginner, Intermediate or Advanced
• Length check to make sure the rider level entered is either 8 or 12 characters
• Type check to make sure the rider level is alphanumeric</td><td>2</td></tr>
</table>

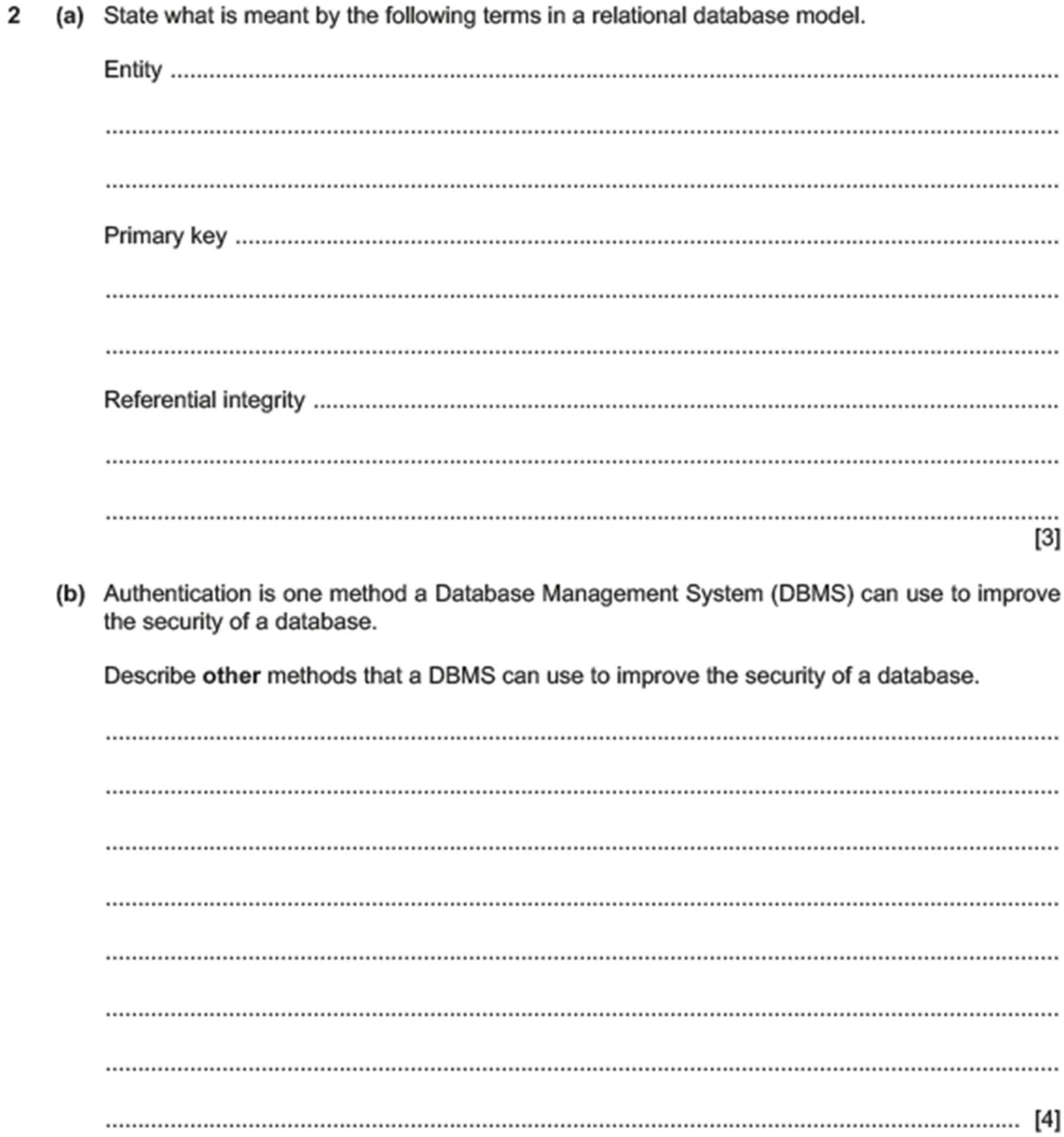


Question	Answer	Marks

2(c)(ii)	**1 mark each**
	• SELECT field Name
	• FROM table HORSE
	• WHERE with Intermediate / Beginner
	• OR with Beginner / Intermediate
	Example answer:
	`SELECT Name`
	`FROM HORSE`
	`WHERE HorseLevel = "Intermediate"`
	`OR HorseLevel = "Beginner";`
		4

2(c)(iii)	**1 mark each** • SUM should be COUNT // SELECT COUNT(STUDENT.RiderLevel) • The WHERE statement needs the table names before each field name // WHERE STUDENT.StudentID = LESSON.StudentID • The OR should be AND // AND Date = #09/09/2023# • Beginner is missing the speech marks // STUDENT.RiderLevel = "Beginner";	**4**

October/November 2023



2 (a) State what is meant by the following terms in a relational database model.

Entity ....

....

....

Primary key ....

....

....

Referential integrity ....

....

.... [3]

(b) Authentication is one method a Database Management System (DBMS) can use to improve the security of a database.

Describe **other** methods that a DBMS can use to improve the security of a database.

....

....

....

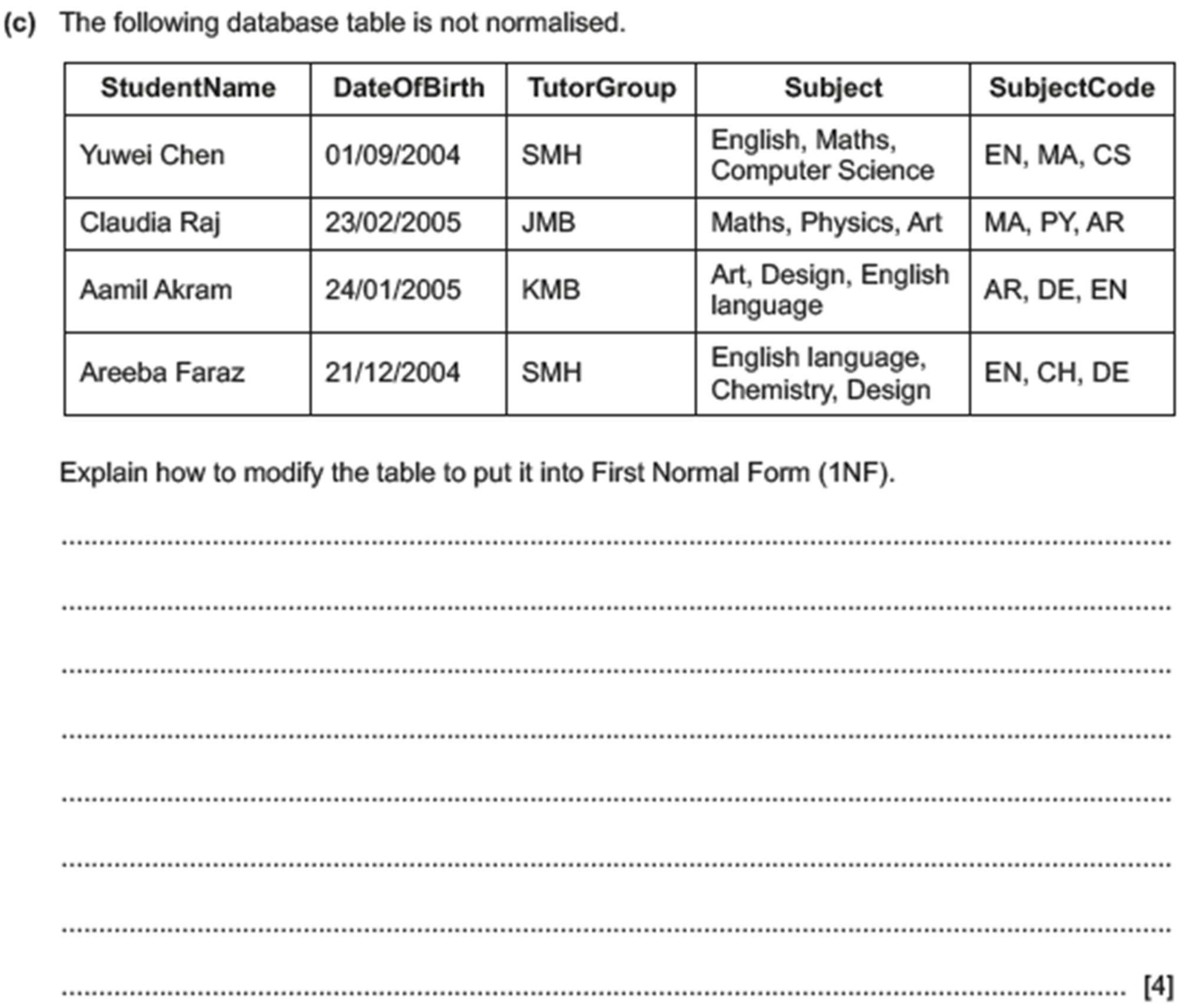
....

....

....

....

.... [4]



(c) The following database table is not normalised.

StudentName	DateOfBirth	TutorGroup	Subject	SubjectCode
Yuwei Chen	01/09/2004	SMH	English, Maths, Computer Science	EN, MA, CS
Claudia Raj	23/02/2005	JMB	Maths, Physics, Art	MA, PY, AR
Aamil Akram	24/01/2005	KMB	Art, Design, English language	AR, DE, EN
Areeba Faraz	21/12/2004	SMH	English language, Chemistry, Design	EN, CH, DE

Explain how to modify the table to put it into First Normal Form (1NF).

....

....

....

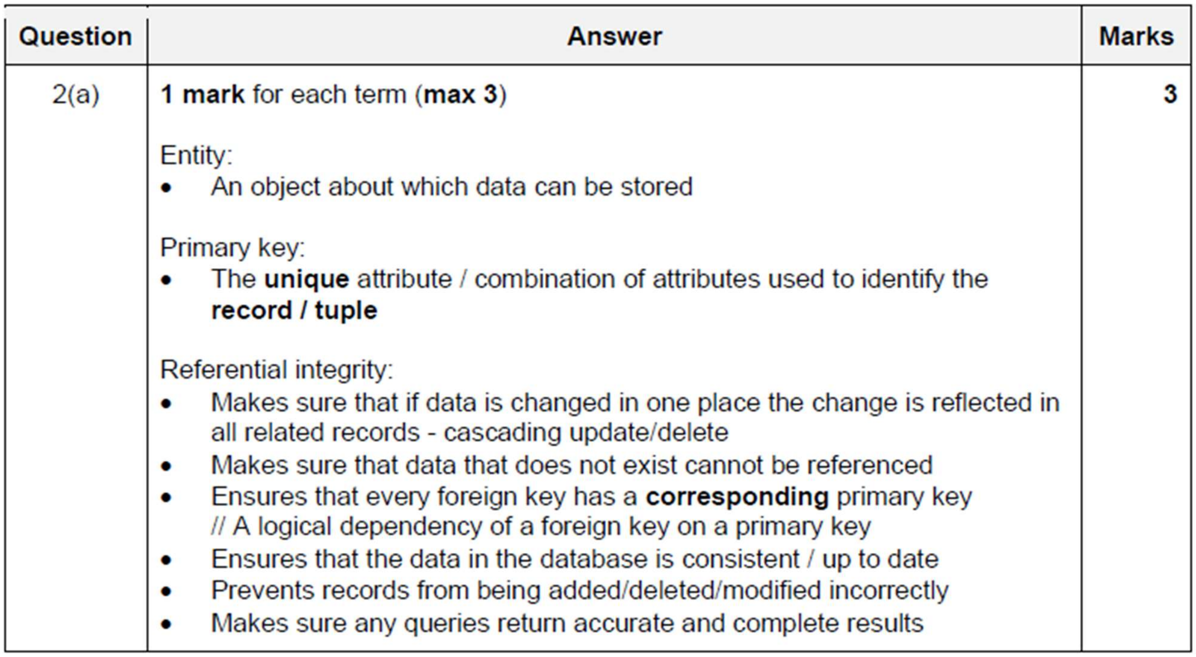
....

....

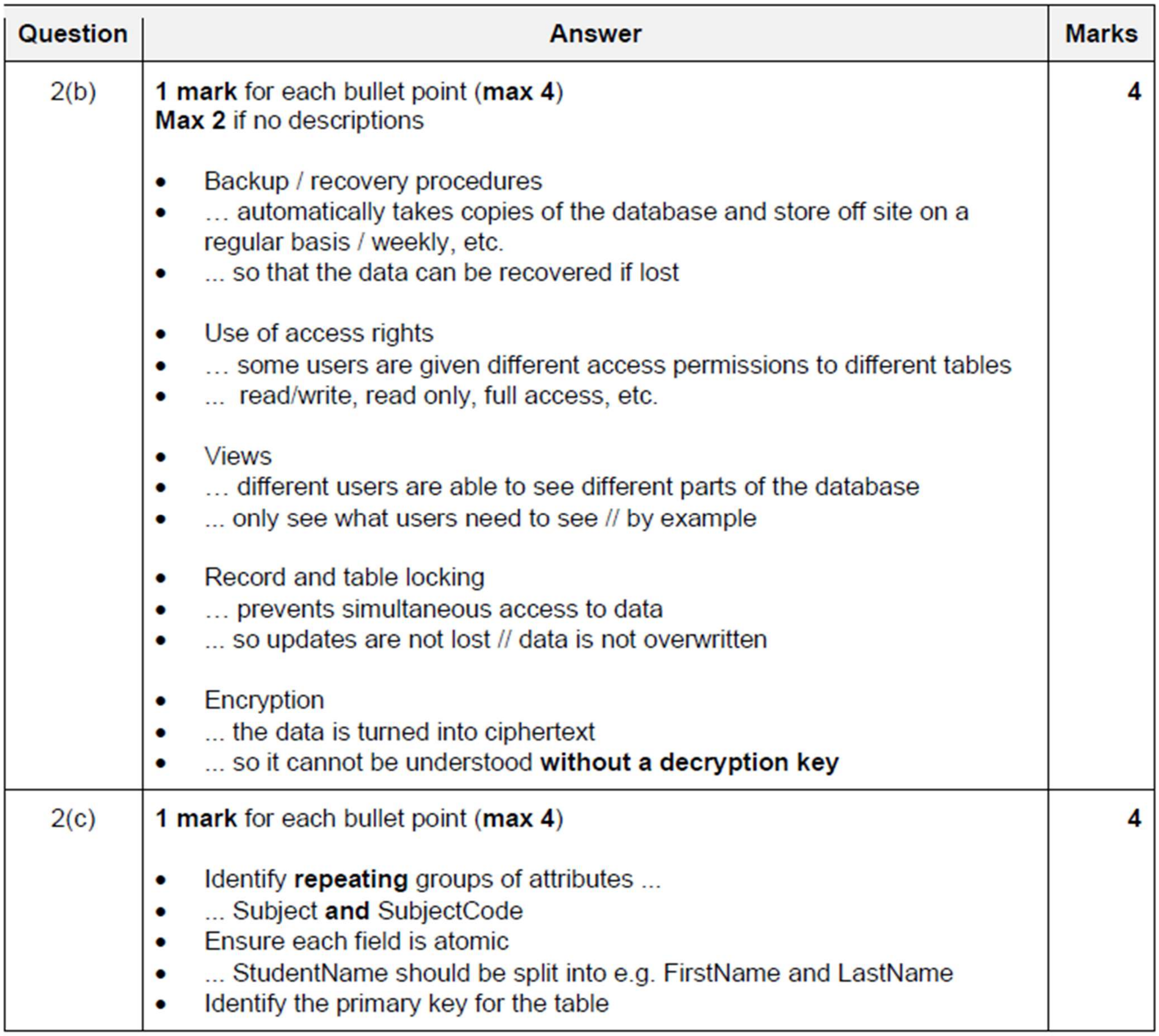
....

....

.... **[4]**



Question	Answer	Marks
2(a)	**1 mark** for each term (**max 3**) Entity: • An object about which data can be stored Primary key: • The **unique** attribute / combination of attributes used to identify the **record / tuple** Referential integrity: • Makes sure that if data is changed in one place the change is reflected in all related records - cascading update/delete • Makes sure that data that does not exist cannot be referenced • Ensures that every foreign key has a **corresponding** primary key // A logical dependency of a foreign key on a primary key • Ensures that the data in the database is consistent / up to date • Prevents records from being added/deleted/modified incorrectly • Makes sure any queries return accurate and complete results	**3**



Question	Answer	Marks
2(b)	**1 mark** for each bullet point (**max 4**) **Max 2** if no descriptions • Backup / recovery procedures • … automatically takes copies of the database and store off site on a regular basis / weekly, etc. • ... so that the data can be recovered if lost • Use of access rights • … some users are given different access permissions to different tables • ... read/write, read only, full access, etc. • Views • … different users are able to see different parts of the database • ... only see what users need to see // by example • Record and table locking • … prevents simultaneous access to data • ... so updates are not lost // data is not overwritten • Encryption • ... the data is turned into ciphertext • ... so it cannot be understood **without a decryption key**	**4**
2(c)	**1 mark** for each bullet point (**max 4**) • Identify **repeating** groups of attributes ... • ... Subject **and** SubjectCode • Ensure each field is atomic • ... StudentName should be split into e.g. FirstName and LastName • Identify the primary key for the table	**4**

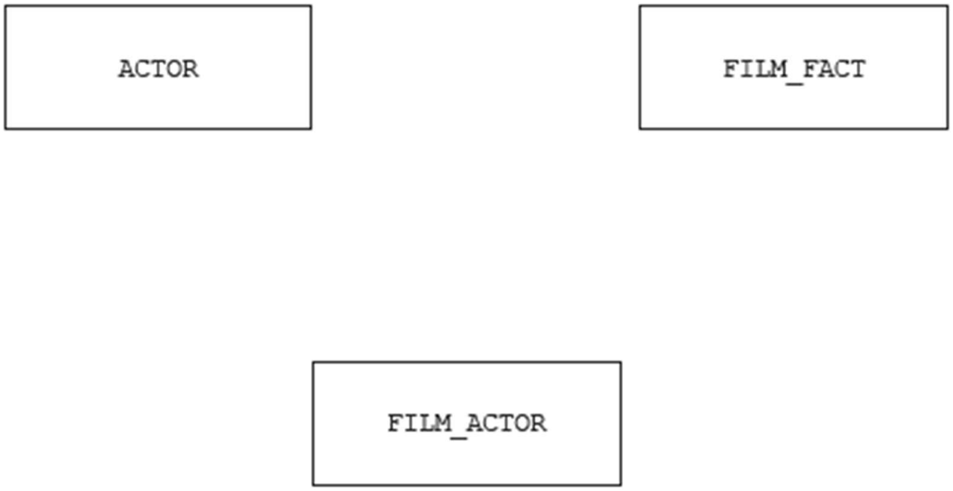
May/June 2022

5 A database, FILMS, stores information about films and actors.

Part of the database is shown:

```
ACTOR(ActorID, FirstName, LastName, DateOfBirth)
FILM_FACT(FilmID, FilmTitle, ReleaseDate, Category)
FILM_ACTOR(ActorID, FilmID)
```

(a) Complete the entity-relationship (E-R) diagram.



[2]

(b) A composite primary key consists of two or more attributes that together form the primary key.

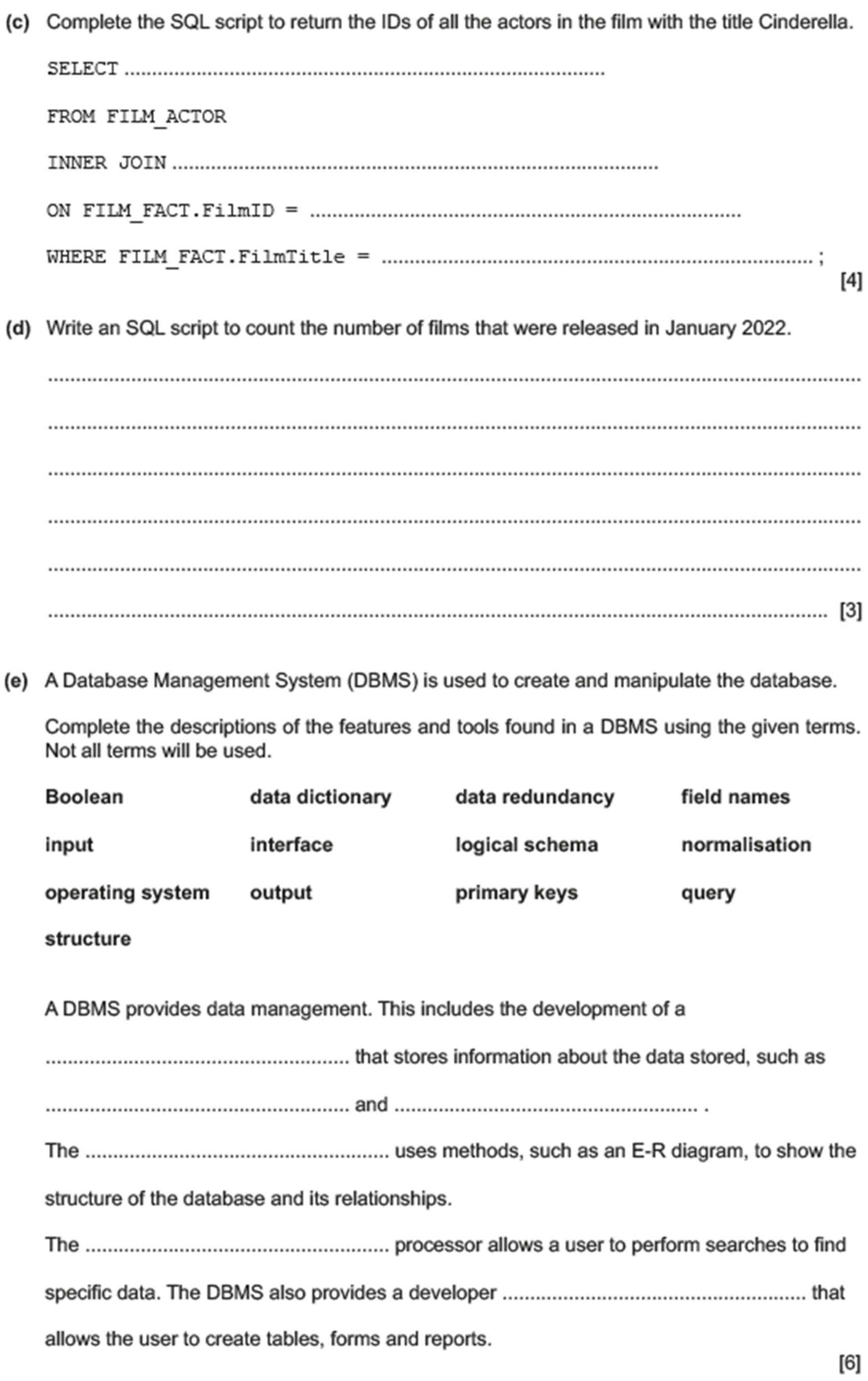
Explain why the table FILM_ACTOR has a composite primary key.

....

....

....

.... [2]



(c) Complete the SQL script to return the IDs of all the actors in the film with the title Cinderella.

```
SELECT ..............................................................................

FROM FILM_ACTOR

INNER JOIN ..........................................................................

ON FILM_FACT.FilmID = ...............................................................

WHERE FILM_FACT.FilmTitle = ........................................................ ;
```

[4]

(d) Write an SQL script to count the number of films that were released in January 2022.

....

....

....

....

....

.... **[3]**

(e) A Database Management System (DBMS) is used to create and manipulate the database.

Complete the descriptions of the features and tools found in a DBMS using the given terms. Not all terms will be used.

Boolean **data dictionary** **data redundancy** **field names**

input **interface** **logical schema** **normalisation**

operating system **output** **primary keys** **query**

structure

A DBMS provides data management. This includes the development of a

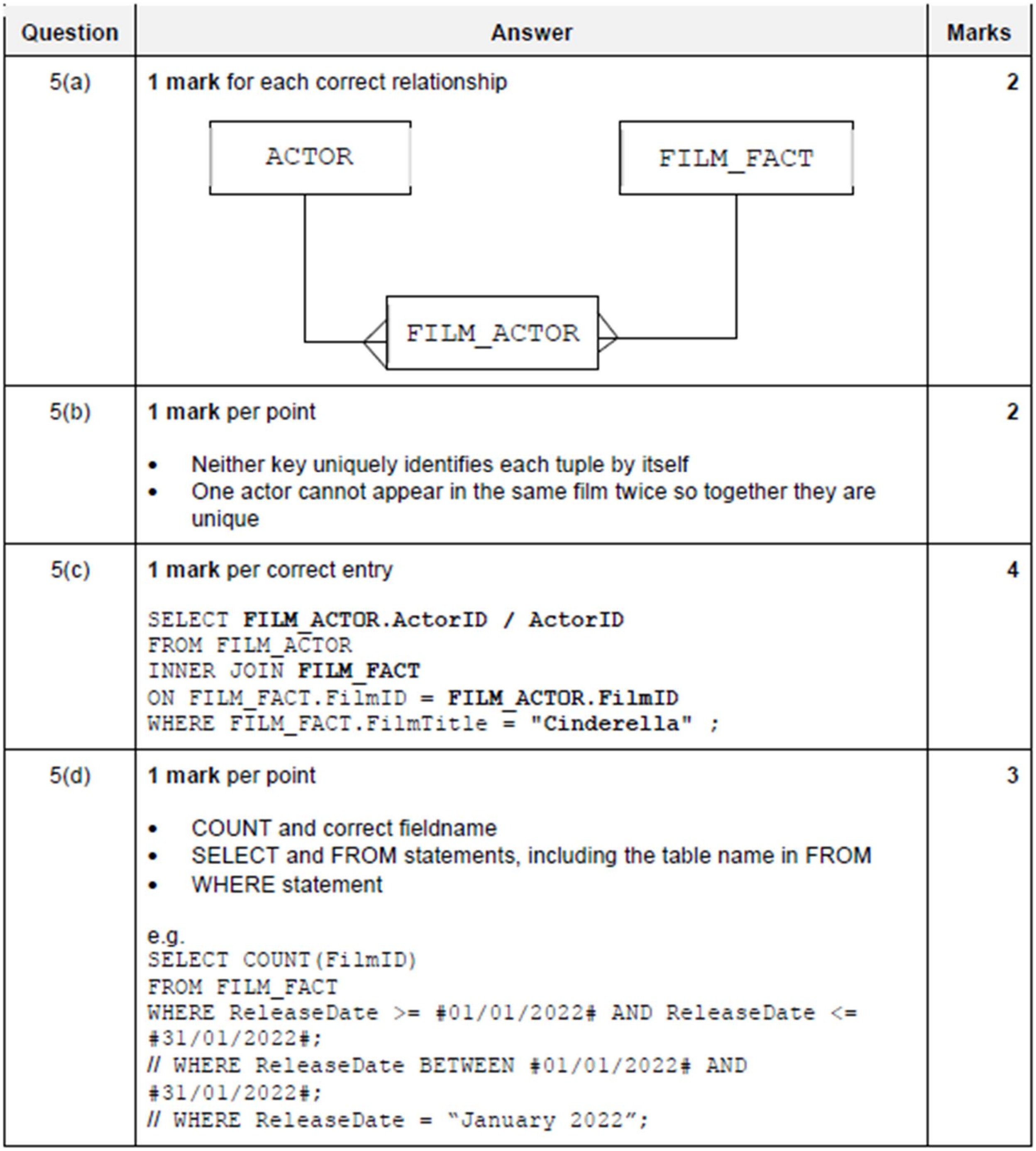
.... that stores information about the data stored, such as

.... and ....

The .... uses methods, such as an E-R diagram, to show the structure of the database and its relationships.

The .... processor allows a user to perform searches to find specific data. The DBMS also provides a developer .... that allows the user to create tables, forms and reports.

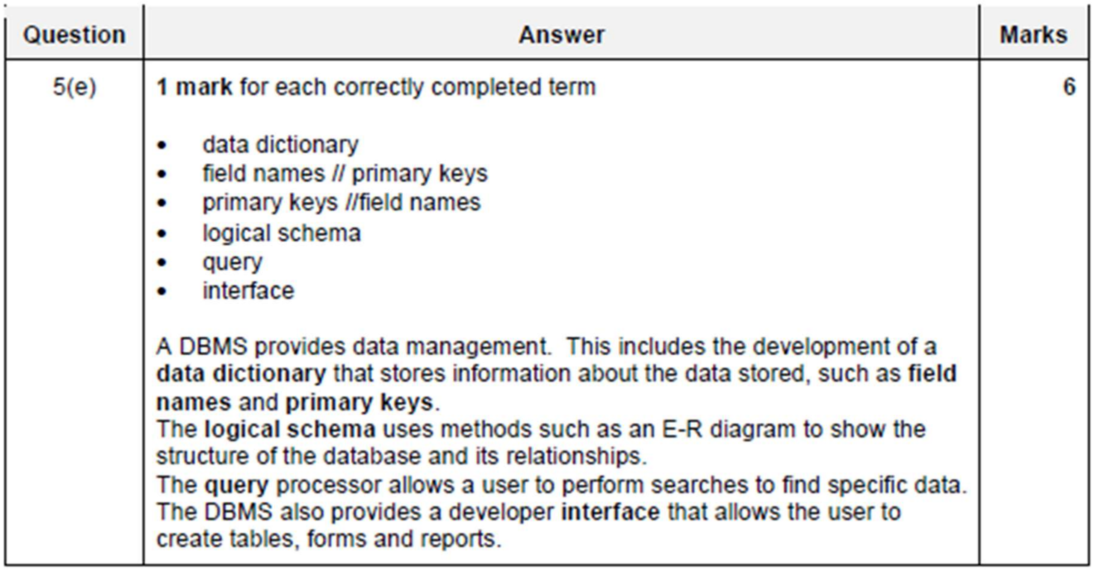
[6]



Question	Answer	Marks
5(a)	**1 mark** for each correct relationship ACTOR FILM_FACT FILM_ACTOR	**2**
5(b)	**1 mark** per point • Neither key uniquely identifies each tuple by itself • One actor cannot appear in the same film twice so together they are unique	**2**

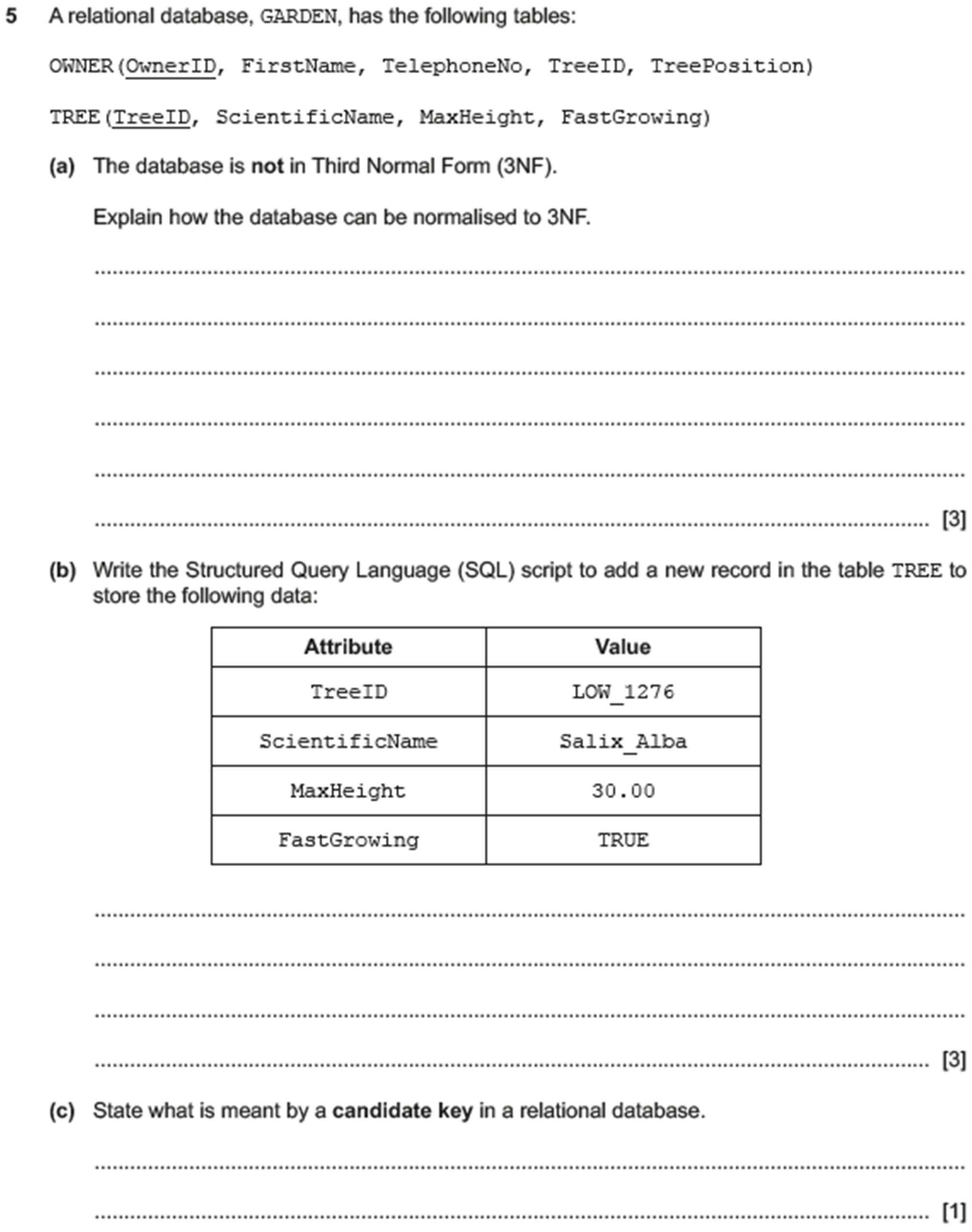
5(c)	**1 mark** per correct entry
	`SELECT` **`FILM_ACTOR.ActorID / ActorID`**
	`FROM FILM_ACTOR`
	`INNER JOIN` **`FILM_FACT`**
	`ON FILM_FACT.FilmID =` **`FILM_ACTOR.FilmID`**
	`WHERE FILM_FACT.FilmTitle =` **`"Cinderella"`** `;`
		4

5(d)	**1 mark** per point
	• COUNT and correct fieldname
	• SELECT and FROM statements, including the table name in FROM
	• WHERE statement
	
	e.g.
	`SELECT COUNT(FilmID)`
	`FROM FILM_FACT`
	`WHERE ReleaseDate >= #01/01/2022# AND ReleaseDate <= #31/01/2022#;`
	`// WHERE ReleaseDate BETWEEN #01/01/2022# AND #31/01/2022#;`
	`// WHERE ReleaseDate = “January 2022”;`
		3



Question	Answer	Marks
5(e)	**1 mark** for each correctly completed term • data dictionary • field names // primary keys • primary keys //field names • logical schema • query • interface A DBMS provides data management. This includes the development of a **data dictionary** that stores information about the data stored, such as **field names** and **primary keys**. The **logical schema** uses methods such as an E-R diagram to show the structure of the database and its relationships. The **query** processor allows a user to perform searches to find specific data. The DBMS also provides a developer **interface** that allows the user to create tables, forms and reports.	6

October/November 2022



5 A relational database, GARDEN, has the following tables:

OWNER(OwnerID, FirstName, TelephoneNo, TreeID, TreePosition)

TREE(TreeID, ScientificName, MaxHeight, FastGrowing)

(a) The database is **not** in Third Normal Form (3NF).

Explain how the database can be normalised to 3NF.

....

....

....

....

....

.... [3]

(b) Write the Structured Query Language (SQL) script to add a new record in the table TREE to store the following data:

Attribute	Value
TreeID	LOW_1276
ScientificName	Salix_Alba
MaxHeight	30.00
FastGrowing	TRUE

....

....

....

.... [3]

(c) State what is meant by a **candidate key** in a relational database.

....

.... [1]

(d) (i) Describe, using an example, what is meant by a **data dictionary**.

....

....

....

.... [2]

(ii) Describe what is meant by a **logical schema**.

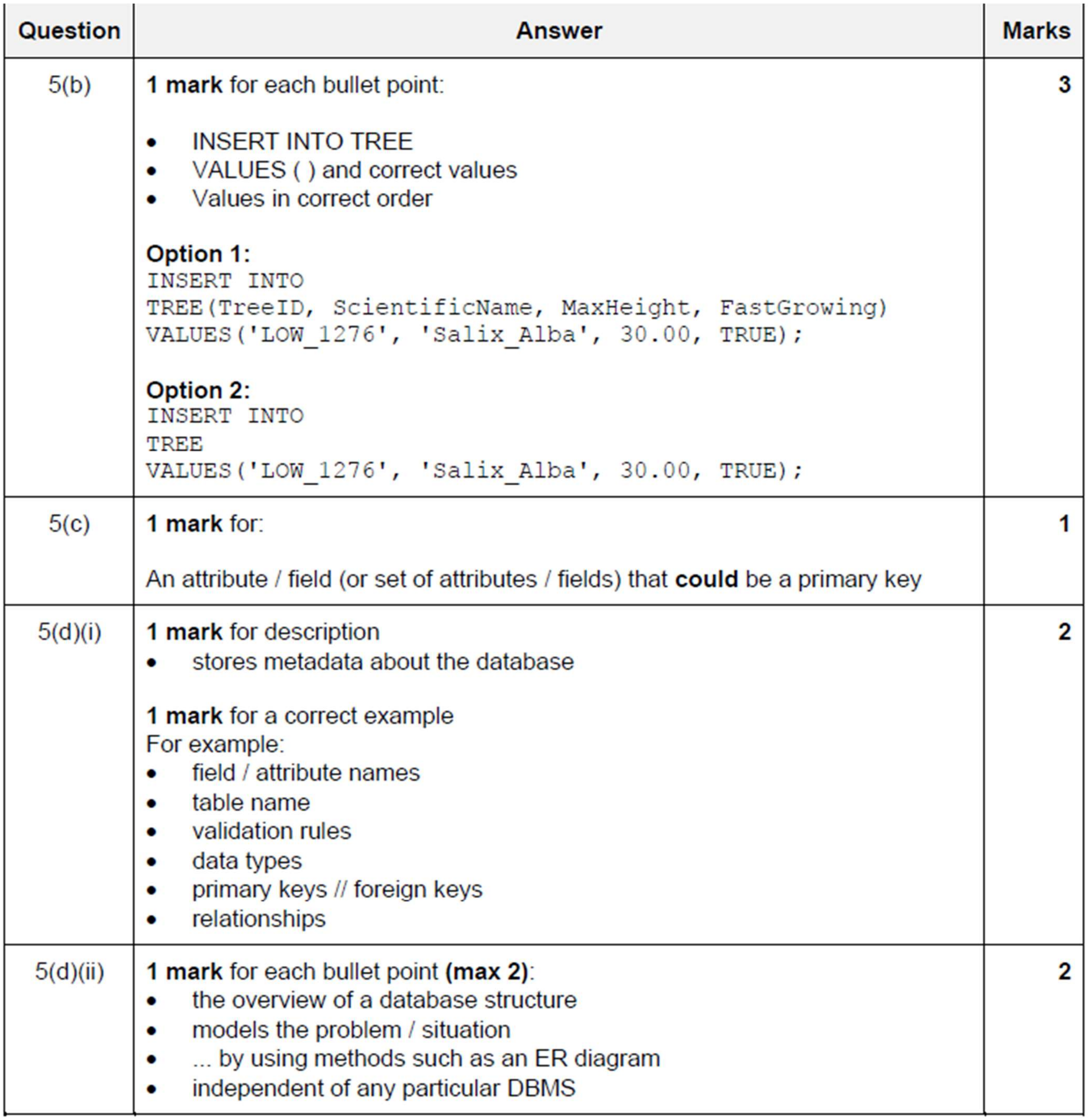
....

....

....

.... [2]

Question	Answer	Marks
5(a)	**1 mark** for each bullet point **(max 3)**: Solution 1: • removing the many-to-many relationship between Owner and Tree • ... by removing TreeID and TreePosition from the Owner table • ... and creating a linking table between Owner and Tree • ... that contains OwnerID, TreeID and TreePosition • ... (composite) primary key of the linking table should be OwnerID and TreeID // insert a named new primary key in the linking table Solution 2: • removing the many-to-many relationship between Owner and Tree • move TreePosition into TREE table • ... put OwnerID into TREE table • create a new table with suitable name (for the species of tree) • ... containing ScientificName, MaxHeight and FastGrowing • ... with ScientificName as primary key // or another suitable primary key	**3**

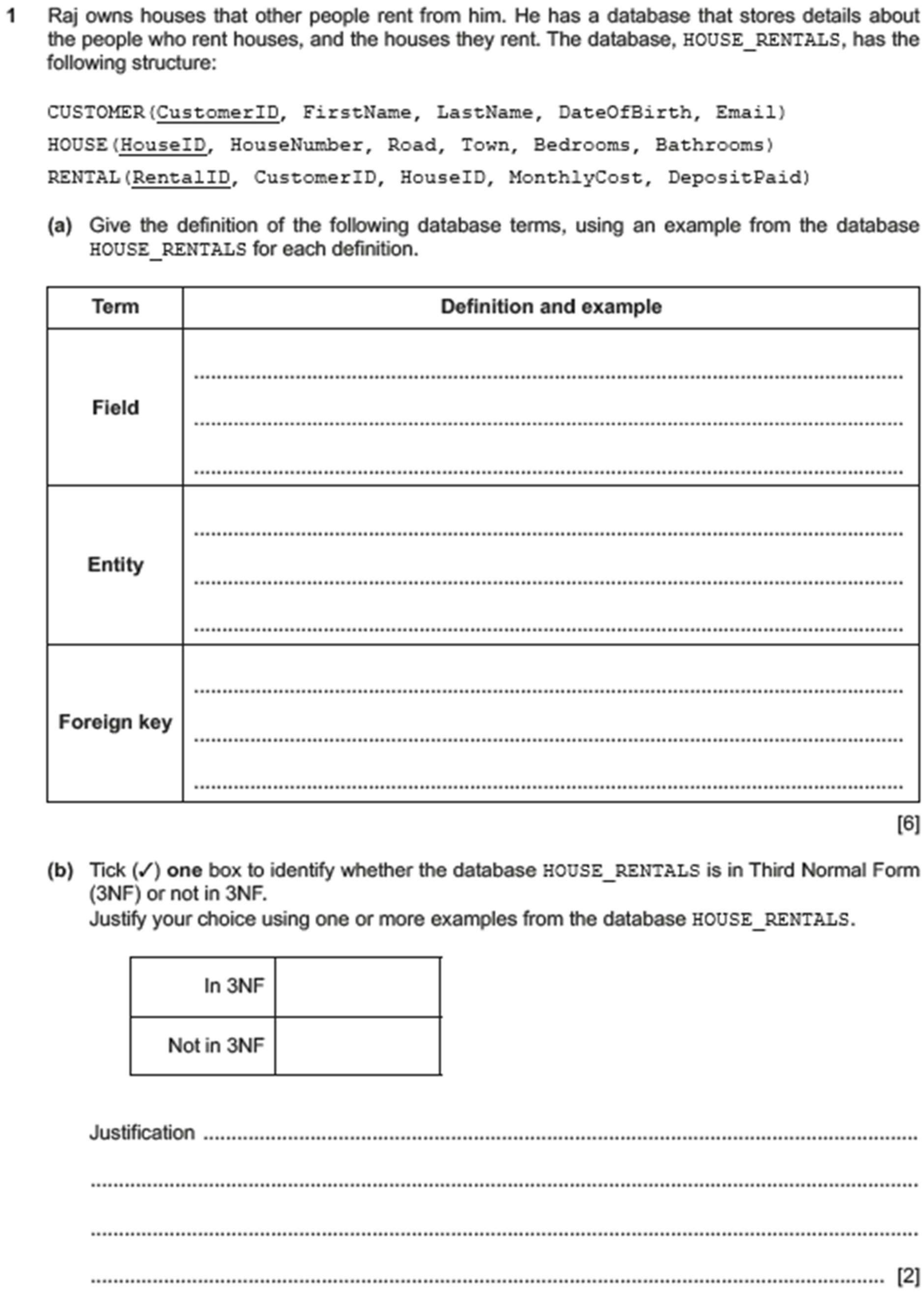


Question	Answer	Marks

5(b)	**1 mark** for each bullet point:
	• INSERT INTO TREE
	• VALUES ( ) and correct values
	• Values in correct order
	Option 1:
	`INSERT INTO`
	`TREE(TreeID, ScientificName, MaxHeight, FastGrowing)`
	`VALUES('LOW_1276', 'Salix_Alba', 30.00, TRUE);`
	Option 2:
	`INSERT INTO`
	`TREE`
	`VALUES('LOW_1276', 'Salix_Alba', 30.00, TRUE);`
		3

5(c)	**1 mark** for: An attribute / field (or set of attributes / fields) that **could** be a primary key	**1**
5(d)(i)	**1 mark** for description • stores metadata about the database **1 mark** for a correct example For example: • field / attribute names • table name • validation rules • data types • primary keys // foreign keys • relationships	**2**
5(d)(ii)	**1 mark** for each bullet point **(max 2)**: • the overview of a database structure • models the problem / situation • ... by using methods such as an ER diagram • independent of any particular DBMS	**2**

May/June 2021



1 Raj owns houses that other people rent from him. He has a database that stores details about the people who rent houses, and the houses they rent. The database, HOUSE_RENTALS, has the following structure:

```
CUSTOMER(CustomerID, FirstName, LastName, DateOfBirth, Email)
HOUSE(HouseID, HouseNumber, Road, Town, Bedrooms, Bathrooms)
RENTAL(RentalID, CustomerID, HouseID, MonthlyCost, DepositPaid)
```

(a) Give the definition of the following database terms, using an example from the database HOUSE_RENTALS for each definition.

Term	Definition and example
Field	....
Entity	....
Foreign key	....

[6]

(b) Tick (✓) **one** box to identify whether the database HOUSE_RENTALS is in Third Normal Form (3NF) or not in 3NF.
Justify your choice using one or more examples from the database HOUSE_RENTALS.

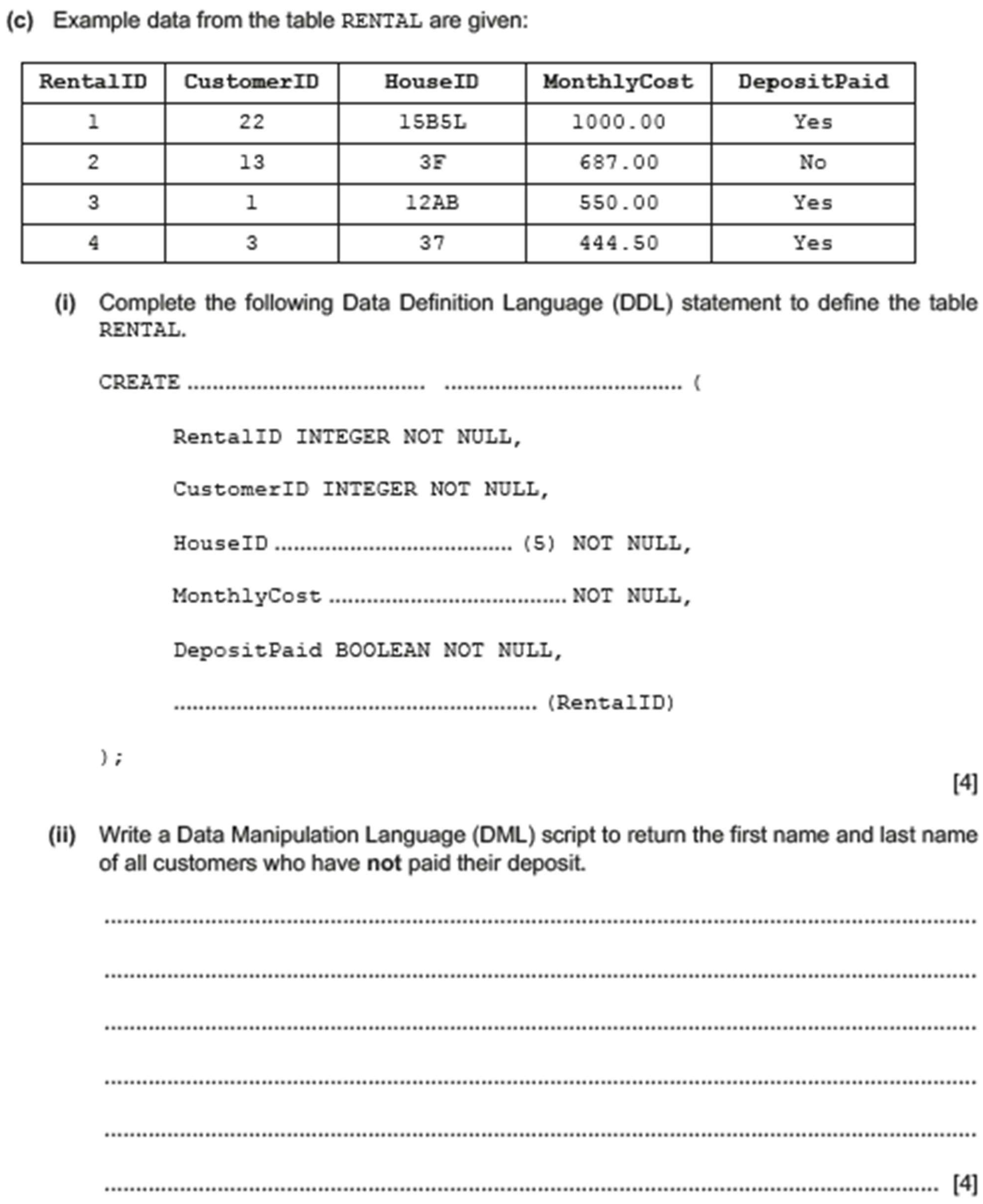
In 3NF	
Not in 3NF	

Justification ....

....

....

.... [2]



(c) Example data from the table RENTAL are given:

RentalID	CustomerID	HouseID	MonthlyCost	DepositPaid
1	22	15B5L	1000.00	Yes
2	13	3F	687.00	No
3	1	12AB	550.00	Yes
4	3	37	444.50	Yes

(i) Complete the following Data Definition Language (DDL) statement to define the table RENTAL.

```
CREATE .................................... .................................... (

       RentalID INTEGER NOT NULL,

       CustomerID INTEGER NOT NULL,

       HouseID .................................... (5) NOT NULL,

       MonthlyCost .................................... NOT NULL,

       DepositPaid BOOLEAN NOT NULL,

       ........................................................ (RentalID)

);
```

[4]

(ii) Write a Data Manipulation Language (DML) script to return the first name and last name of all customers who have **not** paid their deposit.

....

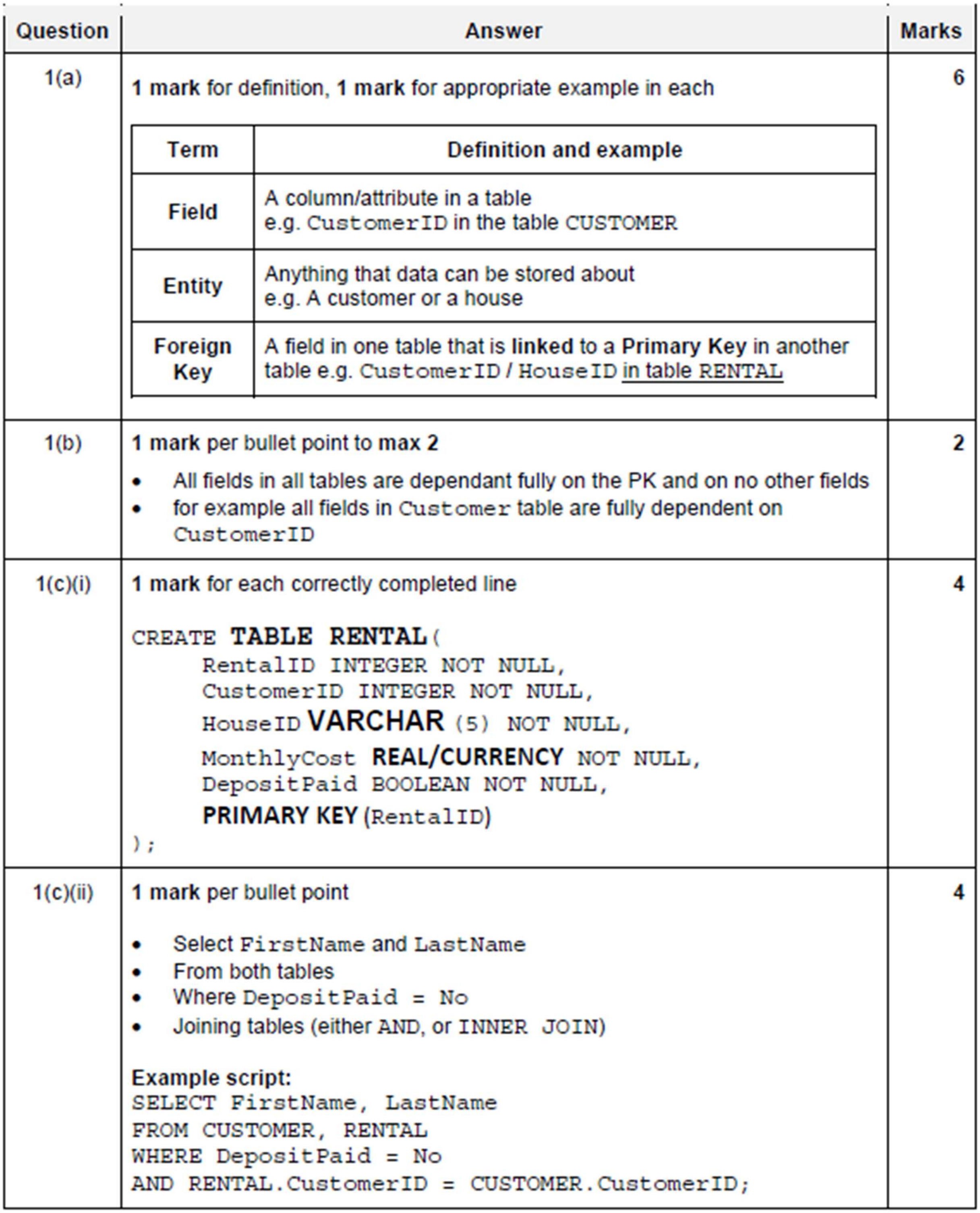
....

....

....

....

.... [4]



<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>1(a)</td><td>

1 mark for definition, **1 mark** for appropriate example in each

<table>
<tr><th>Term</th><th>Definition and example</th></tr>
<tr><td><b>Field</b></td><td>A column/attribute in a table
e.g. CustomerID in the table CUSTOMER</td></tr>
<tr><td><b>Entity</b></td><td>Anything that data can be stored about
e.g. A customer or a house</td></tr>
<tr><td><b>Foreign Key</b></td><td>A field in one table that is <b>linked</b> to a <b>Primary Key</b> in another table e.g. CustomerID / HouseID in table RENTAL</td></tr>
</table>

</td><td>6</td></tr>
<tr><td>1(b)</td><td>

1 mark per bullet point to **max 2**

- All fields in all tables are dependant fully on the PK and on no other fields
- for example all fields in Customer table are fully dependent on CustomerID

</td><td>2</td></tr>
<tr><td>1(c)(i)</td><td>

1 mark for each correctly completed line

```
CREATE TABLE RENTAL(
     RentalID INTEGER NOT NULL,
     CustomerID INTEGER NOT NULL,
     HouseID VARCHAR (5) NOT NULL,
     MonthlyCost REAL/CURRENCY NOT NULL,
     DepositPaid BOOLEAN NOT NULL,
     PRIMARY KEY (RentalID)
);
```

</td><td>4</td></tr>
<tr><td>1(c)(ii)</td><td>

1 mark per bullet point

- Select FirstName and LastName
- From both tables
- Where DepositPaid = No
- Joining tables (either AND, or INNER JOIN)

Example script:

```
SELECT FirstName, LastName
FROM CUSTOMER, RENTAL
WHERE DepositPaid = No
AND RENTAL.CustomerID = CUSTOMER.CustomerID;
```

</td><td>4</td></tr>
</table>

October/November 2021

6 A shop sells plants to customers. The shop manager has a relational database to keep track of the sales.

The database, PLANTSALES, has the following structure:

PLANT(PlantName, QuantityInStock, Cost)

CUSTOMER(CustomerID, FirstName, LastName, Address, Email)

PURCHASE(PurchaseID, CustomerID)

PURCHASE_ITEM(PurchaseID, PlantName, Quantity)

(a) The database is normalised.

(i) The table lists the following three stages of normalisation:

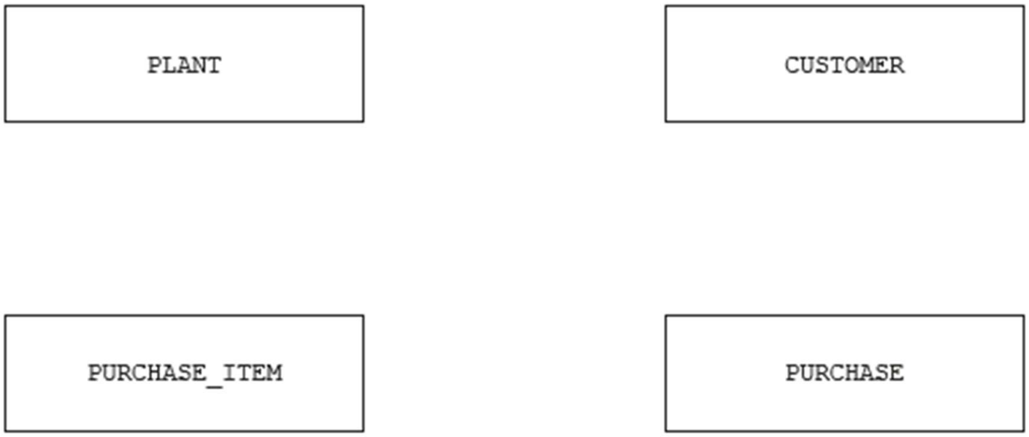
- The first stage is from a database that is not normalised (0NF) to First Normal Form (1NF).
- The second stage is from 1NF to Second Normal Form (2NF).
- The third stage is from 2NF to Third Normal Form (3NF).

Tick (✓) **one** box in each row to identify the appropriate stage for each task.

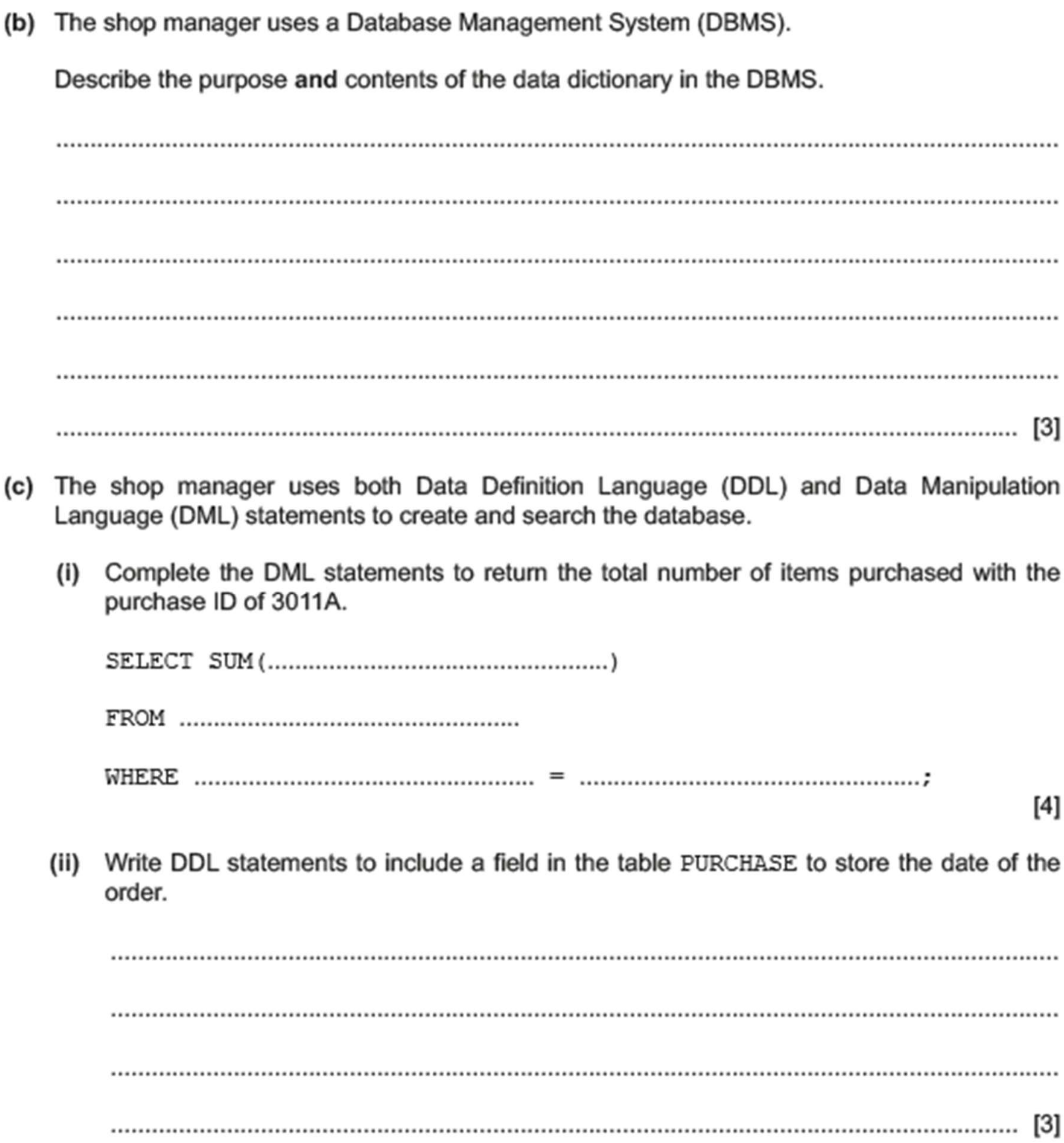
Task	**Normalisation stage**		
	0NF to 1NF	**1NF to 2NF**	**2NF to 3NF**
Remove any partial key dependencies			
Remove any repeating groups of attributes			
Remove any non-key dependencies			

[2]

(ii) Draw an entity-relationship (E-R) diagram for the database PLANTSALES.



[3]



(b) The shop manager uses a Database Management System (DBMS).

Describe the purpose **and** contents of the data dictionary in the DBMS.

....

....

....

....

....

.... [3]

(c) The shop manager uses both Data Definition Language (DDL) and Data Manipulation Language (DML) statements to create and search the database.

(i) Complete the DML statements to return the total number of items purchased with the purchase ID of 3011A.

```
SELECT SUM(................................................)

FROM ..................................................

WHERE .................................................. = ..................................................;
```

[4]

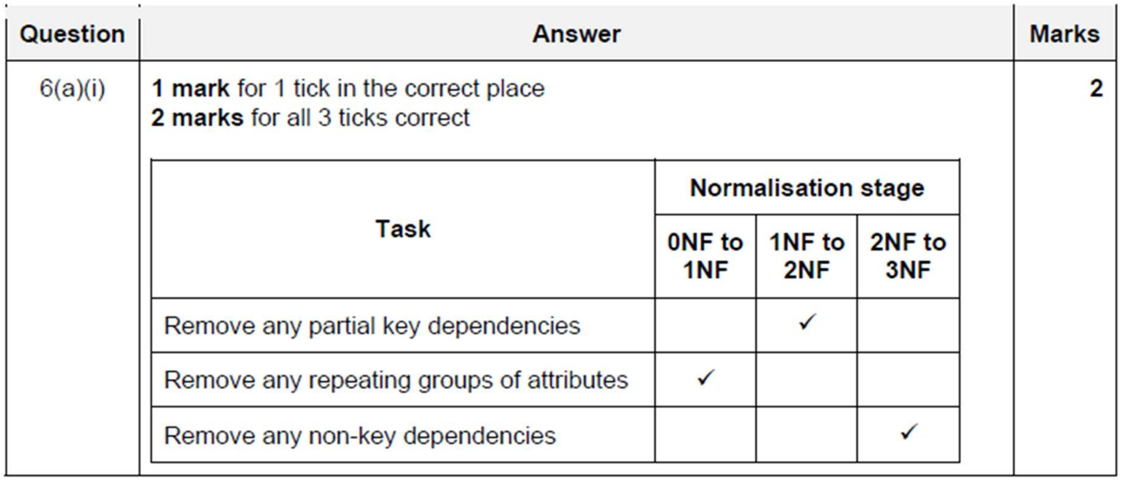
(ii) Write DDL statements to include a field in the table `PURCHASE` to store the date of the order.

....

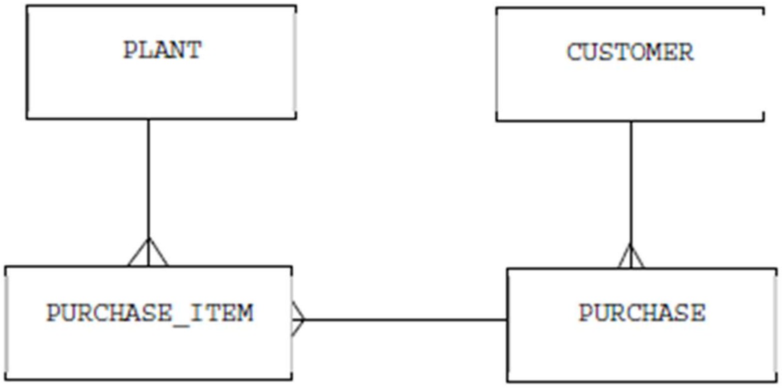
....

....

.... [3]



Question	Answer				Marks
6(a)(i)	**1 mark** for 1 tick in the correct place **2 marks** for all 3 ticks correct				2
	Task	**Normalisation stage**			
		0NF to 1NF	**1NF to 2NF**	**2NF to 3NF**	
	Remove any partial key dependencies		✓		
	Remove any repeating groups of attributes	✓			
	Remove any non-key dependencies			✓	

Question	Answer	Marks
6(a)(ii)	**1 mark** for each correct relationship 	3
6(b)	**1 mark** for description of purpose • Stores metadata about the database **1 mark** for each example of contents to **max 2** e.g. • field / attribute names • table name • validation rules • data types • primary keys // foreign keys • relationships	3

6(c)(i)	**1 mark** for each correctly completed space
	
	`SELECT SUM(`**`Quantity`**`)`
	`FROM `**`PURCHASE_ITEM`**
	`WHERE `**`PurchaseID`**` = `**`"3011A"`**`;`
		4

6(c)(ii)	**1 mark** per bullet point
	
	• ALTER TABLE PURCHASE
	• ADD OrderDate
	• Suitable data type, e.g. DATE
	
	`ALTER TABLE PURCHASE`
	`ADD OrderDate DATE;`
		3