Database definitions: 2

Challenge 1

Drag and drop the most appropriate term into each row of the table below to correctly label the definitions. Each term can only be used once.

Definition	Term
A collection of organised data where the data is held in related tables.	
A collection of related data made up of fields and records.	
A collection of data for one item, person, or thing.	
A single piece of information which is stored within a row. For example: first name, email, date of birth.	
A single field that acts as a unique identifier for each record in a table.	
A set of two or more fields that together uniquely identify each record in a table.	
A field in one table that is a primary key in another table, and is used to create a link between these tables.	

Items:

record composite key relational database primary key foreign key table field

Benefits of normalisation

Practice 2

There are many benefits of using a database system to store data. Identify which **three** statements, from those given below, correctly specify benefits of using a well-defined, normalised database.

- [ ] Elimination of duplicated data
- [ ] Elimination of deletion anomalies
- [ ] Elimination of redundant data
- [ ] Elimination of update anomalies

Quiz:
STEM SMART Computer Science Week 44

Entity relationships: 4

Practice 2

Kate is developing an application that will distribute a quiz that her fellow college students can attempt each month. The students will access the monthly quiz via the school portal. For October, the theme of the quiz is Black History Month.

Kate has written down the requirements for her application:

- A quiz consists of a number of unique multiple choice questions
- Each question has at least three choices, only one of them is correct
- A question can appear in only one quiz
- A choice belongs to only one question
- A student can attempt the same quiz more than once

Kate wants to use a relational database to store information about the quizzes. This includes details about the quizzes that have been published, the questions that belong to each quiz, and the choices that are available for each question. Also, the details of the students that attempt each quiz, when they attempted each quiz and what choice they submitted for each of the questions.

The information is modelled using six entities `Quiz`, `Question`, `Choice`, `Student`, `Attempt` and `AttemptAnswer`. The description in standard notation for the entities is as follows:

```
Quiz(QuizId, Name, PublishedDate)
Question(QuestionId, QuizId, Description, QuestionOrder)
Choice(ChoiceId, QuestionId, Description, ChoiceOrder, IsCorrect)
Student(StudentId, FirstName, LastName, YearGroup, Email)
Attempt(AttemptId, StudentId, QuizId, SubmittedDate, SubmittedTime)
AttemptAnswer(AttemptId, ChoiceId)
```

Part A Student - Quiz

Kate wants to produce the entity–relationship diagram for the database. What is the relationship between the entities `Student` and `Quiz`?

- ○ One-to-many
- ○ Many-to-many
- ○ One-to-one

Part B Student - Attempt

What is the relationship between the `Student` and `Attempt` entities?

- One-to-many
- One-to-one
- Many-to-many

Part C Quiz - Attempt

What is the relationship between the `Quiz` and `Attempt` entities?

- Many-to-many
- One-to-many
- One-to-one

Quiz:
STEM SMART Computer Science Week 44

DB design problem: 1

Challenge 2

Charlie runs a dog training club that offers a range of different training classes:

- Some classes, such as obedience training, run only once a week; other classes, such as puppy training, run several times a week.
- There is only one room available, so only one class can run at any given time.

Dog owners must register their dog(s) before they can make bookings:

- An owner can own more than one dog
- An owner can book a dog into many classes

Charlie has set up a system to manage his business. He has designed a relational database to hold the information that he needs. This includes details about the dog owners, the type of dog(s) that are registered for each owner, the classes that are offered in the club, and the bookings that each owner has made. The initial database design is as follows:

```
Owner (OwnerId, Name, Phone, Email)
Dog (DogName, OwnerId, Breed, Colour)
Class (Day, Time, ClassType, MaxPlaces)
Booking (OwnerId, ClassType, DogName)
```

Charlie tries to use his new system, but encounters some problems. Because of the **incorrect database design**, there are some tasks that he is unable to do.

Select the **three** tasks from the list that Charlie **can** do with the current system design.

- [ ] Sort his reports by the surname of the owner.
- [ ] Register many dogs with the same owner.
- [ ] Add two owners with the same name.
- [ ] Book a dog into the same class the following week.
- [ ] Target emails at owners of a specific breed of dog.
- [ ] Add two dogs with the same name.

Quiz:
STEM SMART Computer Science Week 44

Improve database design

Challenge 2

Kate is developing an application that will distribute a quiz that her fellow college students can attempt each month. The students will access the quiz via the school portal. For October, the theme of the quiz is Black History Month.

Kate has written down the requirements for her application:

- A quiz consists of a number of unique multiple choice questions
- Each question has at least three choices, only one of them is correct
- A question can appear in only one quiz
- A choice belongs to only one question
- A student can attempt the same quiz more than once

Kate wants to use a relational database to store information about the quizzes. This includes details about the quizzes that have been published, the questions that belong to each quiz, and the choices that are available for each question. Also, the details of the students that participate in each quiz, when they attempted each quiz and what choice they submitted for each of the questions. The information is modelled using six entities `Quiz`, `Question`, `Choice`, `Student`, `Attempt` and `AttemptAnswer`. The description in standard notation for the entities is as follows:

```
Quiz(QuizId, Name, PublishedDate)
Question(QuestionId, QuizId, Description, QuestionOrder)
Choice(ChoiceId, QuestionId, Description, ChoiceOrder, IsCorrect)
Student(StudentId, FirstName, LastName, YearGroup, Email)
Attempt(AttemptId, StudentId, QuizId, SubmittedDate, SubmittedTime)
AttemptAnswer(AttemptId, ChoiceId)
```

Each entity is implemented in the database using a table. Kate's friend Diane looks at the table definitions and suggests that Kate can simplify her database by replacing the tables `Attempt` and `AttemptAnswers` with the following table:

```
QuizAttempt(StudentId, QuizId, SubmittedDate, SubmittedTime, ChoiceId)
```

Does Diane's suggestion improve the design of the database?

- Yes, using `StudentId, QuizId, SubmittedDate, SubmittedTime` as the composite primary key for table `QuizAttempt` ensures that the composite primary key values are unique, as only one quiz attempt can be recorded at a specific date and time.
- No, the `QuizAttempt` table with `StudentId, QuizId, SubmittedDate, SubmittedTime` as the composite primary key results in a table where the same values of the composite primary key repeat for all the choices that are submitted as part of the same quiz attempt.
- No, the primary key of `QuizAttempt` should be `StudentId, QuizId`. The table `QuizAttempt` is a link table between the tables `Student` and `Quiz` and so the correct composite primary key is the combination of their primary keys.

INSERT query 3

Challenge 2

Kate is developing an application that will distribute a monthly quiz to her fellow college students. The students will access the quiz via the school portal. For October, the theme of the quiz is Black History Month.

The full list of requirements for her application is below:

- A quiz consists of a number of unique multiple choice questions
- Each question has at least three choices, only one of them is correct
- A question can appear in only one quiz
- Once a student submits a quiz, their attempt is recorded in the database and their answers can't be changed
- A student can attempt a quiz more than once

The table definitions for the database are:

```
Quiz(QuizId, Name, PublishedDate)
Question(QuestionId, QuizId, Description, QuestionOrder)
Choice(ChoiceId, QuestionId, Description, ChoiceOrder, IsCorrect)
Student(StudentId, FirstName, LastName, YearGroup, Email)
Attempt(AttemptId, StudentId, QuizId, SubmittedDate, SubmittedTime)
AttemptAnswer(AttemptId, ChoiceId)
```

Kate wants to create the records for the Black History Month quiz in the database. So far she has come up with the quiz name and the description for the first question:

Quiz Name: Black History Month
Question 1: What is Mae Jemison famous for?

Select the correct partial SQL statements and drag them into the correct order to create the records for the quiz and question as above. You only need to use six partial statements.

Available items

```
INSERT INTO Question
```

```
VALUES (5,'Black History Month',01/10/2021);
```

```
VALUES (1,5,'What is Mae Jemison famous for?',1);
```

```
(QuestionId, QuizId, Description, QuestionOrder)
```

```
INSERT INTO Quiz
```

```
VALUES (1,5,'What is Mae Jemison famous for?','Q1');
```

```
(QuizId, Name, PublishedDate)
```

```
VALUES (5,'Black History Month','01/10/2021');
```

UPDATE query 1

Practice 1

Charlie runs a dog training club that offers a range of different training classes:

- Some classes, such as obedience training, run only once a week; other classes, such as puppy training, run several times a week.
- There is only one room available, so only one class can run at any given time.

Dog owners must register their dog(s) before they can make bookings:

- An owner can own more than one dog
- An owner can book a dog into many classes

Charlie has set up a system to manage his business. He has designed a relational database to hold the information that he needs. This includes details about the dog owners, the type of dog(s) that are registered for each owner, the classes that are offered in the club, and the bookings that each owner has made. The initial database design is as follows:

```
Owner (OwnerId, Name, Phone, Email)
Dog (DogName, OwnerId, Breed, Colour)
Class (Day, Time, ClassType, MaxPlaces)
Booking (OwnerId, ClassType, DogName)
```

Because of high demand for agility training, Charlie has employed an assistant to help with the classes. This means that he can increase the maximum capacity of the classes from 10 to 15 dogs. There are two types of agility class: basic and advanced.

Charlie is going to use SQL to update the database accordingly. Which of the following statements is the one that he needs?

- ○ UPDATE Class SET MaxPlaces = 15;
- ○ UPDATE Class SET MaxPlaces = 15 WHERE MaxPlaces = 10;
- ○ UPDATE Class SET MaxPlaces = 15 WHERE (ClassType = “basic agility” OR “advanced agility”) AND MaxPlaces = 10;
- ○ UPDATE Class SET MaxPlaces = 15 WHERE ClassType = “basic agility” OR ClassType = “advanced agility”;

Quiz:
STEM SMART Computer Science Week 44

CREATE TABLE: 2

Challenge 2

Kate is developing an application that will distribute a monthly quiz to her fellow college students. The students will access the quiz via the school portal. For October, the theme of the quiz is Black History Month.

The full list of requirements for her application is below:

- A quiz consists of a number of unique multiple choice questions, displayed in the order specified
- Each question has at least three choices, only one of them is correct
- A question can appear in only one quiz
- Once a student submits a quiz, their attempt is recorded in the database and their answers can't be changed
- A student can attempt a quiz more than once

The table definitions for the database are:

```
Quiz(QuizId, Name, PublishedDate)
Question(QuestionId, QuizId, Description, QuestionOrder)
Choice(ChoiceId, QuestionId, Description, ChoiceOrder, IsCorrect)
Student(StudentId, FirstName, LastName, YearGroup, Email)
Attempt(AttemptId, StudentId, QuizId, SubmittedDate, SubmittedTime)
AttemptAnswer(AttemptId, ChoiceId)
```

Put the following partial SQL statements in order to create the `Question` table in the database. You only need to use four partial statements.

Available items

```
  Description VARCHAR(100),
  QuestionOrder BOOLEAN,
```

```
  PRIMARY KEY (QuestionId));
```

```
CREATE TABLE Question (
```

```
  PRIMARY KEY (QuestionId)
  FOREIGN KEY (QuizId) REFERENCES Quiz(QuizId));
```

```
  QuestionId INTEGER,
  QuizId INTEGER,
```

```
  Description VARCHAR(100),
  QuestionOrder INTEGER,
```

Clinic: delete customer

Practice 1

Repair & Reform is a muscle therapy clinic that provides support for everyday pain and sports injuries. The company uses a relational database to hold the details of appointments, treatments, customers, and staff.

You can study the table layouts and see some sample data on the Repair & Reform database information page.

A sample of the records from the `customer` table is shown below.

customer_id	first_name	last_name	email	telephone
1	Kohen	Ceyhun	kohen106@example.com	0770090048
2	Vaclovas	Derry	vaclovas167@example.net	0770090015
3	Mar	Sylvia	mar160@example.edu	0770090049

The customer whose ID is 2 has moved away and will no longer be able to attend their scheduled appointments. These appointments must be deleted from the database.

The SQL statement shown below has a placeholder. You must replace this with a statement that will delete all the records from the `appointment` table for the relevant customer. Below the placeholder is a `SELECT` query that counts the total number of appointments. **Do not delete the SELECT statement** as you need this to answer the question.

Reload the page to restore the original state of the question and the database if you accidentally delete the wrong data.

SQL

```
1  -- Write delete statement here --
2
3
4  SELECT COUNT(*) AS total_appointments
5  FROM appointment;
```

Enter the **total number of appointments** that remain in the database once the appointments for the customer with ID number 2 have been deleted.

Clinic: insert, update staff

Challenge 1

Repair & Reform is a muscle therapy clinic that provides support for everyday pain and sports injuries. The company uses a relational database to hold the details of appointments, treatments, customers, and staff.

You can study the table layouts and see some sample data on the Repair & Reform database information page.

The database currently contains records for appointments made between January 2nd and 31st, 2024 (inclusive). A sample of the data in the `appointment` table is shown below.

customer_id	treatment_ref	appointment_date	start_time	staff_id	discount
21	UB_arm	2024-01-02	09:00:00	3	0.25
29	LB_foot	2024-01-02	09:30:00	1	0.2
36	LB_thigh	2024-01-02	11:00:00	3	0

Part A Update staff

Sondra, one of the staff members, needs to swap their afternoon shifts on the 30th and 31st of January 2024. Deepti has agreed to take Sondra's shifts from 1pm on these days.

Sondra's staff ID is 1 and Deepti's staff ID is 2.

The SQL statement shown below has a placeholder. You need to replace this with a statement to `UPDATE` the appointment table so that the staff ID of the affected appointments is updated.

Below the placeholder is a `SELECT` statement that counts the total number of appointments recorded for Deepti. **Do not delete this statement** as you need to run this to answer the question.

Reload the page to restore the original state of the question and the database if you accidentally update the wrong data.

When querying this database, you need to use single quotation marks around a date value or a time value. Dates are in the format `YYYY-MM-DD` and times are in 24-hour format `hh:mm:ss`

SQL

```sql
-- Write update statement here --


SELECT COUNT(*) AS num_appointments
FROM appointment
WHERE staff_id = 2;
```

Enter the **total number of appointments Deepti (with staff ID 2) has in the database** once the affected appointments have been updated.

Part B Insert appointment

Deepti has received a new appointment request from a customer for an arm and elbow treatment. The full details of the appointment are shown in the table below.

Field name	Data type	Data to insert
`customer_id`	Integer	17
`treatment_ref`	Text	UB_arm
`appointment_date`	Date	2024-01-30
`start_time`	Time	16:15:00
`staff_id`	Integer	2
`discount`	Real	0.25

Edit the SQL statement so that the new appointment is added to the `appointment` table. **Do not delete the SELECT statement** as you need this to answer the question.

SQL

```
1  -- Write insert statement here --
2
3
4  SELECT AVG(discount)
5  FROM appointment;
```

Calculate the **average discount** of all the appointments once the new appointment has been added. Reload the page to restore the original state of the database if you accidentally insert the wrong data.

Type your answer rounded to **three decimal places**.