Task 1: database design and development (part A)

Anytime Flowers is a florist that tailor-makes bunches of flowers for customers.

When a customer comes into the shop, an employee notes down the customer's details, their order information and the price of the order. They give each order an order code.

Customers select one type of flower (rose, lily, tulip or daffodil), the size of the bunch of flowers (small, medium or large) and whether or not they would like chocolates with the flowers. They also select a date for collecting their order. Customers can include a message with their flowers.

1a Anytime Flowers wants to create a database to store customer and order details.

Complete the order details in the analysis of inputs table below:

(2 marks)

Customer details:	**Order details:**
Customer ID Name Address Telephone number	1 mark for identifying: ♦ order ID/order code 1 mark for identifying: ♦ order price ♦ collection date ♦ flower type ♦ size of bunch ♦ chocolates ♦ message

- Check your answers carefully, as you cannot return to part A after you hand it in.
- When you are ready, hand part A to your teacher or lecturer and collect part B.

Candidate name_____ Candidate number_____

Task 1: database design and development (part B)

1b Your teacher or lecturer will provide you with a database file containing two linked tables.

Using the data dictionary below, complete the relational database by:

- identifying two fields where the validation shown below has yet to be applied
- adding the validation to the two identified fields

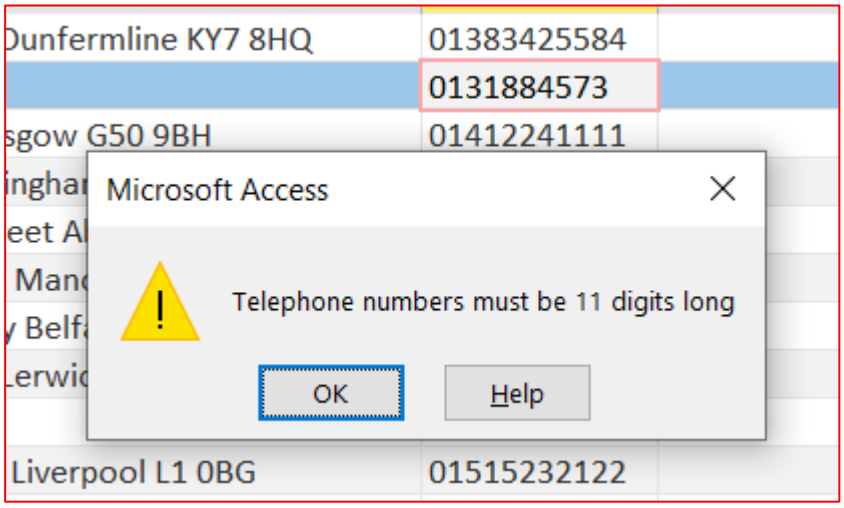
(2 marks)

Entity: Customer					
Attribute name	**Key**	**Type**	**Size**	**Required**	**Validation**
customerID	PK	number		Y	
forename		text	40	Y	
surname		text	50	Y	
address		text	100	N	
telephoneNo		text	11	N	Length = 11

Entity: FlowerOrder					
Attribute name	**Key**	**Type**	**Size**	**Required**	**Validation**
orderID	PK	text	10	Y	
dateDue		date		Y	
price		number		Y	Range: >= 5.00 and <= 50.00
flowerType		text	8	Y	Restricted choice: rose, lily, tulip, daffodil
bunchSize		text	6	Y	Restricted choice: small, medium, large
chocolates		Boolean		Y	
message		text	200	N	
customerID	FK	number		Y	Existing customerID from Customer table

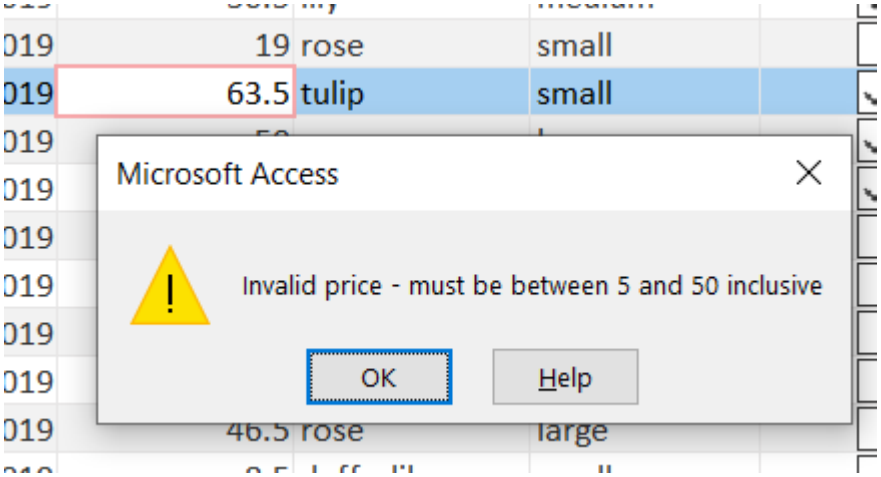
Print evidence to show that you have added the validation to the database to match the data dictionary requirements.

Telephone Length Check



Field Size	11
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	Len([telephoneNo])=11
Validation Text	Telephone numbers must be 11 digits long
Required	No
Allow Zero Length	Yes
Indexed	No
Unicode Compression	Yes
IME Mode	No Control
IME Sentence Mode	None
Text Align	General

Price Range Check



Field Size	Single
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	0
Validation Rule	>=5 And <=50
Validation Text	Invalid price - must be between 5 and 50 inclusive
Required	Yes
Indexed	No

1c (i) A customer would like to change their order from ‘rose’ to ‘tulip’. The price of the order will change from £34 to £17. The orderID is CHQ3848.

Implement **one** SQL statement that will make the required changes to the order.

(4 marks)

Print evidence of the SQL statement and the FlowerOrder table, clearly showing that the changes have been implemented.

```
UPDATE FlowerOrder
SET flowerType = "tulip", price = 17
WHERE orderID = "CHQ3848";
```

```
UPDATE FlowerOrder
SET flowerType = "tulip", price = 17
WHERE orderID="CHQ3848";
```

BUD15136	30/03/2019	14	lily	small			544
CHQ3848	28/03/2019	17	tulip	medium		Thanks for everythin	392
DEI246272	30/03/2019	27	lily	medium			228

(ii) A new customer provides their name and telephone number.

Implement an SQL statement that will add their details to the database.

Name: Richard Glass
Telephone number: 07654029336

(2 marks)

Assign them customerID — 2986.

Print evidence of the SQL statement and the Customer table, clearly showing that the changes have been implemented.

```
INSERT INTO Customer (customerID, forename, surname, telephoneNo)
VALUES ('2986', 'Richard', 'Glass', '07654 029336');
```

```
INSERT INTO Customer ( customerID, forename, surname, telephoneNo )
VALUES ('2986', 'Richard', 'Glass', '07654 029336');
```

2986	Richard	Glass		07654 02933

1d Anytime Flowers wants to find the names of all customers who had placed orders for the smallest bunch of flowers.

The following incorrect SQL statement is written.

```
SELECT customerName
FROM Customer, FlowerOrder
WHERE size = "smallest"
AND Customer.customerID = FlowerOrder.customerID;
```

Test this SQL statement.

State two reasons why this SQL statement failed.

(2 marks)

Reason 1 **Any 2 from the following** • no field called “customerName” • no field called “size” • restricted choice field doesn’t include “smallest” option
Reason 2

Candidate name_____ Candidate number_____