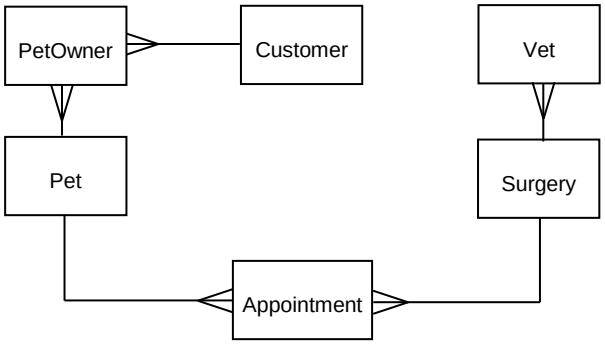
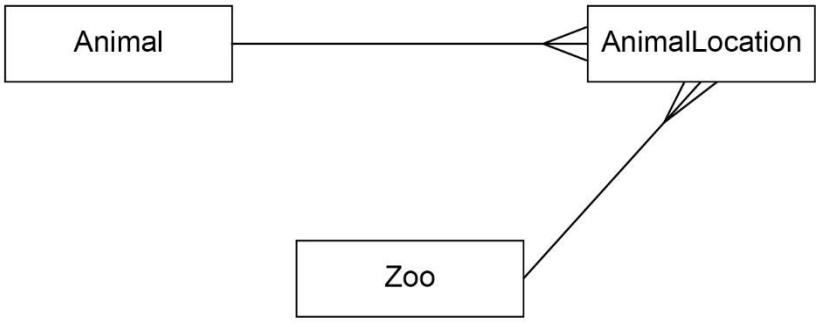
1	**1**	**All marks AO2 (analysis)**  **1 mark:** Entity Appointment added and correct relationships and degree drawn to Pet and Surgery entities. **1 mark:** Entity Customer added and correct relationship and degree drawn to PetOwner. **Note:** If PetOwner relation not created then allow this mark if a many-to-many relationship is drawn between Pet and Customer, even though this is not fully normalised. **1 mark:** Entity PetOwner added and correct relationship and degree drawn to Pet. **A.** entity names do not have to match diagrams exactly but must convey same purpose. **Ignore the inclusion of any additional entities and the drawing of any other relationships, whether they are correct or not.**	**3**

Question			Marks
2	1	**Mark is AO1 (understanding)** **C** The primary key in each relation consists of only one attribute; **R.** if more than one lozenge shaded	**1**

Question			Marks
2	2	**All marks AO2 (analyse)**  **1 mark**: one-to-many relationship between Animal and AnimalLocation **1 mark:** one-to-many relationship between Zoo and AnimalLocation If no marks awarded then award **1 mark** for many-to-many relationship between Animal and Zoo **Max 1** if any incorrect relationships drawn (ignore the inclusion of a many-to-many relationship between Animal and Zoo)	**2**

Question			Marks

2	3	**All marks AO3 (programming)**
		
		`AnimalID INT PRIMARY KEY,`
		//
		`AnimalID INT,`
		`PRIMARY KEY(AnimalID),`
		
		`IndividualName VARCHAR(50),`
		`Species VARCHAR(40),`
		`DateOfBirth DATE,`
		`Sex VARCHAR(6)`
		
		These **are AO3 marks so syntax must be correct** (including commas) to award them
		
		1 mark: `AnimalID`, with sensible data type and identified as primary key
		
		1 mark: two other fields with sensible data types and lengths (if given)
		
		1 mark: two other fields with sensible data types and lengths (if given)
		
		DPT. data type before fieldname (**Note: penalisation is of marks not mistakes**)
		DPT. incorrect punctuation - missing commas, unnecessary semi-colons, brackets etc but ignore bracket or semi-colon added at very end
		A. Any sensible types. Lengths do not need to be specified
		I. Case
		
		Valid alternative SQL types are:
		
		• Alternative types for `AnimalID`: tinyint, smallint, mediumint, integer, number, byte
		• Alternative types for `IndividualName`, `Species` and `Sex`: char, nchar, nvarchar, ntext, longvarchar, varchar2, nvarchar2, text, tinytext, mediumtext, longtext, string
		• Alternative types for `DateOfBirth`: datetime, datetime2, datetimeoffset, smalldatetime **R.** time
			3

Question			Marks

2	4	**5 marks for AO2 (analyse) and 2 marks for AO3 (programming)**
		
		Mark Scheme
		
		AO2 (analyse) – 5 marks:
		
		1 mark for correctly analysing the data model and identifying the tables that data needs to be extracted from (`Animal`, `AnimalLocation`) and the fields that need to be extracted (`IndividualName`, `DateArrived`), and including these and no other tables or fields in the query
		A. inclusion of unnecessary table `Zoo` as long as it is correctly linked to the `AnimalLocation` table by a linking condition
		
		1 mark for correctly identifying the condition to select the correct species of animal: `Species = "Red Panda"` or correctly identifying the condition to select the correct zoo: `ZooName = "Ashdale Park"`
		
		1 mark for correctly identifying the condition to link the two tables: `Animal.AnimalID = AnimalLocation.AnimalID` - see example 3 for how to apply this to nested solutions.
		R. do not award mark if additional linking conditions for tables that the query does not use are included
		
		1 mark for at least one pair of conditions that would identify some animals that were at the zoo during the required period, or **2 marks** for conditions that would identify all animals that were at the zoo during the period. Example conditions (not the only ones) that would do this are:
		
		Example full set of conditions 1 – award 2 marks for all conditions or 1 mark for any pair of conditions that would identify some animals at the zoo
		
		`DateArrived < "01/04/2020" AND DateLeft > "31/05/2020"`
		(animal arrived before and left after time period)
		
		`DateArrived <= "31/05/2020" AND DateLeft = "01/01/0001"`
		(animal arrived before end of time period and has not left)
		
		`DateArrived >= "01/04/2020" AND DateArrived <= "31/05/2020"`
		(animal arrived during the time period)
		
		`DateLeft >= "01/04/2020" AND DateLeft <= "31/05/2020"`
		(animal left during the time period)
		
		Example full set of conditions 2 – award 1 mark for the DateArrived condition and either of the DateLeft conditions or 2 marks for all three conditions
		
		`DateArrived <= "31/05/2020" AND (DateLeft >= "01/04/2020" OR DateLeft = "01/01/0001")`
		(animal arrived before end of time period and left after start of time period or has not left)
			7

Example incomplete conditions – award 1 mark for pair of conditions

`DateArrived >= "01/04/2020" AND DateLeft <= "31/05/2020"`
(animal arrived and left during the time period)

Note: Award a **maximum of 2 of the 4 available marks** for the correct conditions if they are not joined by the correct logical operators.

Note: The AO2 marks for analysing the data model should be awarded regardless of whether correct SQL syntax is used or not as they are for data modelling, not syntactically correct SQL programming.

A. mark(s) can be awarded for the correct logical conditions even if the required tables are not identified as being used by the query
A. > instead of >= and < instead of <=
A. ≥, ≤, => and =<

AO3 (programming) – 2 marks:

1 mark for fully correct SQL in two of the three clauses (SELECT, FROM, WHERE)

OR

2 marks for fully correct SQL in all three clauses (SELECT, FROM, WHERE)

Note:
- For the SELECT and FROM SQL clauses to count as correct SQL, they must have the correct field and table names in them.
- For the WHERE clause to count as correct it must include at least one correct condition, but does not have to include them all (ignore missing conditions or irrelevant conditions), however the whole WHERE clause must have correct syntax.

A. table names before fieldnames separated by a full stop
A. use of `Alias`/`AS` command eg `FROM AnimalLocation AS AL` then use of `AL` as the table name and note that command `AS` is not required eg `FROM AnimalLocation AL`
A. `INNER JOIN` written as one word ie `INNERJOIN`
A. insertion of spaces into fieldnames
I. unnecessary brackets so long as they would not stop the query working
A. use of any type of quotation marks, hashes or no delimiters around dates and times
A. > instead of >= and < instead of <=
R. ≥, ≤, => and =<
I. inclusion of an ORDER BY clause
DPT. for unnecessary punctuation – allow one semicolon at the very end of the statement, but not at the end of each clause
DPT. for fieldname before table name

Overall Max 6 if solution does not work fully

Example Solutions

Example 1 – All conditions in WHERE clause

```
SELECT IndividualName, DateArrived
FROM Animal, AnimalLocation
WHERE Species = "Red Panda"
  AND ZooName = "Ashdale Park"
  AND Animal.AnimalID = AnimalLocation.AnimalID
  AND
  ( DateArrived < "01/04/2020" AND DateLeft > "31/05/2020"
    OR DateArrived <= "31/05/2020" AND DateLeft = "01/01/0001"
    OR DateArrived >= "01/04/2020" AND DateArrived <=
       "31/05/2020"
    OR DateLeft >= "01/04/2020" AND DateLeft <= "31/05/2020" )
```

Example 2 – Use of INNER JOIN

```
SELECT IndividualName, DateArrived
FROM Animal INNER JOIN AnimalLocation ON
  Animal.AnimalID = AnimalLocation.AnimalID
WHERE Species = "Red Panda"
  AND ZooName = "Ashdale Park"
  AND
  ( DateArrived < "01/04/2020" AND DateLeft > "31/05/2020"
    OR DateArrived <= "31/05/2020" AND DateLeft = "01/01/0001"
    OR DateArrived >= "01/04/2020" AND DateArrived <=
       "31/05/2020"
    OR DateLeft >= "01/04/2020" AND DateLeft <= "31/05/2020" )
```

Example 3 – A Nested Solution

```
SELECT IndividualName, DateArrived
FROM (SELECT AnimalID, IndividualName
      FROM Animal
      WHERE Species = "Red Panda"
     ) AS RP INNER JOIN AnimalLocation
       ON RP.AnimalID = AnimalLocation.AnimalID
WHERE ZooName = "Ashdale Park"
  AND
  ( DateArrived < "01/04/2020" AND DateLeft > "31/05/2020"
    OR DateArrived <= "31/05/2020" AND DateLeft = "01/01/0001"
    OR DateArrived >= "01/04/2020" AND DateArrived <=
       "31/05/2020"
    OR DateLeft >= "01/04/2020" AND DateLeft <= "31/05/2020" )
```

Refer nested solutions to team leaders for marking

Question			Marks
2	5	**All marks AO2 (analyse)** **Advantage (Max 1):** It will be quicker to lookup an animal's current location; The current location of an animal can be identified without having to query/search the AnimalLocation relation // only the Animal/one relation needs to be searched to identify the location of an animal // the current location of an animal can be identified with a less complex query/search; **NE.** easier to lookup an animal's current location **R.** it will be possible to identify an animal's current location **Disadvantage (Max 1):** Additional storage space will be required; This will introduce data redundancy (as the information can already be found from the AnimalLocation relation); Data inconsistency could occur (as the current location in the Animal relation might not match the current location in the AnimalLocation relation); More updates will be required when an animal is moved between zoos; **A.** the database will no longer be normalised	**2**

Qu	Pt	Marking guidance	Total marks
3	1	**Mark is AO2 (analyse)** **B**; (Each product is only supplied by one supplier) **R.** if more than one lozenge shaded	1