1	1	**All marks AO2 (analysis)** Customer(CustomerID, Forename, Surname, TelephoneNumber) Appointment(PetID, Date, Time, SurgeryName) PetOwner(CustomerID, PetID) **1 mark:** Customer relation created and contains the correct attributes and CustomerID identified as the entity identifier. **I.** any additional reasonable attributes, including PetID if the PetOwner relation has not been created, **but** reject PetID included if the PetOwner entity has been created. If PetID is included then it is acceptable for it to be in or not in the entity identifier. **1 mark:** Appointment relation created and contains the correct attributes. **I.** any additional reasonable attributes including CustomerID, VetID **A.** date and Time given as one combined field. **1 mark:** Composite entity identifier of PetID, Date and Time identified for Appointment relation. **A.** creation of a new attribute to be the entity identifier eg AppointmentID. **A.** date and Time given as one combined field. **1 mark:** PetOwner relation created and contains the correct attributes and no others. Additionally, either: • both attributes identified as a composite entity identifier, or, • **A.** a new entity identifier eg OwnershipID created. **R.** Just one of PetID or CustomerID given as entity identifier. **For all mark points** **A.** alternative names for relations and attributes created by candidate, so long as meaning is clear. **R.** use of incorrect attribute names for attributes already named in question paper. **A.** spaces in relation and attribute names. **I.** if any unnecessary relations are created. **I.** any representation for foreign keys. **Accept responses written in SQL – ignore syntactical errors and data type errors in such responses.**	**4**

2	**1**	**Mark is for AO2 (analysis)** That the buyer will only view the same property once on a particular day; **R.** each visit made by only one buyer	**1**

Qu	Pt	Marking guidance	Total marks
3	1	**Mark is AO2 (analyse)** **B**; (FacilityID, BookingDate, EndTime) **R.** if more than one lozenge shaded	**1**

Qu	Pt	Marking guidance	Total marks
3	2	**All marks AO2 (analyse)** The design is not normalised // there is (unnecessary) data duplication // there is data redundancy // inconsistent data could occur // (one of the) attributes are determined by attributes that are not (part of) the primary key; If a customer made more than one booking then their details would need to be entered more than once / each time // there would be redundancy in relation to the customer data // customer data could be stored multiple times; If customer details were entered more than once they could be inconsistent // there could be inconsistency in the customer data // updates may need to be made to multiple records if a customer's details changed; **A.** updates to customer details would be harder to perform as **BOD** Deleting all of the bookings that a customer made would also delete the data about the customer; It would not be possible to store details about a customer before they had made a booking; It would be harder to identify all the bookings for one customer (as they did not have a unique identifier) // it may be impossible (**A.** difficult) to distinguish between two customers with the same name (if they did not have an email address); **Notes:** • For all mark points (other than the first) it must be stated that it is the customer data that is the issue to award the mark. • Accept points stated the other way around, ie as advantages of the new design instead of reasons to reject the original design. **Max 2**	**2**

Qu	Pt	Marking guidance	Total marks

3	3	**All marks AO3 (programming)**
		
		`FacilityID INT PRIMARY KEY,`
		//
		`FacilityID INT,`
		`PRIMARY KEY(FacilityID),`
		
		`Description VARCHAR(100),`
		`MaxPeople INT,`
		`PricePerHour SMALLMONEY`
		
		1 mark: `FacilityID`, with sensible data type and identified as primary key.
		
		1 mark: two fields other than the primary key have sensible data types and lengths (if given).
		
		1 mark: fully correct definition, with syntactically correct SQL including commas separating each line of code.
		
		A. any sensible types. Lengths do not need to be specified.
		I. brackets at the start / end of the code
		
		Valid alternative SQL types are:
		
		• Alternative types for `FacilityID` and `MaxPeople`: tinyint, smallint, mediumint, integer, number, byte.
		• Alternative types for `Description`: char, nchar, nvarchar, ntext, longvarchar, varchar2, nvarchar2, text, tinytext, mediumtext, longtext, string.
		• Alternative types for `PricePerHour`: money, float, real, decimal, double, numeric, currency. **R.** integer only types.
			3

Qu	Pt	Marking guidance	Total marks

3	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 (`FacilityForSport, Booking`) and the fields that need to be extracted (`FacilityID, StartTime, EndTime`), and including these and no other tables or fields in the query
		
		1 mark for correctly identifying the condition to select facilities suitable for the correct sport: `Sport = "Basketball"` or correctly identifying the condition to select bookings on the required date: `BookingDate = "15/06/2021"`
		
		1 mark for correctly identifying the condition to link the two tables: `Booking.FacilityID = FacilityForSport.FacilityID`
		
		1 mark for at least one condition that would identify some overlapping bookings and no bookings that don't overlap, or **2 marks** for conditions that would identify all overlapping bookings and no bookings that don't overlap. Example conditions (not the only ones) that would identify all overlapping bookings:
		
		Example set of conditions 1
		
		`StartTime <= "14:15" AND EndTime >= "16:15"`
		(existing booking starts before and ends after new booking)
		`StartTime >= "14:15" AND StartTime <= "16:15"`
		(existing booking starts during new booking)
		`EndTime >= "14:15" AND EndTime <= "16:15"`
		(existing booking ends during new booking)
		
		Example set of conditions 2
		
		`StartTime <= "16:15" AND EndTime >= "14:15"`
		(existing booking starts before or at the same time as the end of new booking and ends after or at the same time as the start of new booking)
		
		Note: Award a **maximum of 2 of the 3 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 <=
			7

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)

Notes:

- For the SELECT clause to count as correct SQL it must have the correct field names in it and no others.
- For the FROM clause to count as correct SQL it must have the correct table names in it. Other, unnecessary tables can also be included so long as they are correctly linked into the query by conditions so would not break it.
- For the WHERE clause to count as correct SQL it must include at least one valid condition, but does not have to include them all.

A. instead of `FacilityForSport.FacilityID` accept `Booking.FacilityID` or just `FacilityID` in the SELECT clause for non-nested queries. For a nested query accept `X.FacilityID` where `X` is the alias of the relation produced by the nested query eg `BookingsAtTime` in example 3.
A. table names before fieldnames separated by a full stop.
A. use of `Alias`/`AS` command eg `FROM Booking AS B` then use of `B` as the table name and note that command `AS` is not required eg `FROM Booking B`.
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. month in date as `6` instead of `06`
A. > instead of >= and < instead of <=
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 FacilityForSport.FacilityID, StartTime, EndTime
FROM FacilityForSport, Booking
WHERE Sport = "Basketball"
  AND Booking.FacilityID = FacilityForSport.FacilityID
  AND BookingDate = "15/06/2021"
  AND
  ( StartTime <= "14:15" AND EndTime >= "16:15"
    OR StartTime >= "14:15" AND StartTime <= "16:15"
    OR EndTime >= "14:15" AND EndTime <= "16:15" )
```

Example 2 – Use of INNER JOIN

```
SELECT FacilityForSport.FacilityID, StartTime, EndTime
FROM FacilityForSport INNER JOIN Booking ON
  Booking.FacilityID = FacilityForSport.FacilityID
WHERE Sport = "Basketball"
  AND BookingDate = "15/06/2021"
  AND
  ( StartTime <= "14:15" AND EndTime >= "16:15"
    OR StartTime >= "14:15" AND StartTime <= "16:15"
    OR EndTime >= "14:15" AND EndTime <= "16:15" )
```

Example 3 – A Nested Solution

```
SELECT FacilityForSport.FacilityID, StartTime, EndTime
FROM ( SELECT FacilityID, StartTime, EndTime
       FROM Booking
       WHERE BookingDate = "15/06/2021"
         AND
         ( StartTime <= "14:15" AND EndTime >= "16:15"
           OR StartTime >= "14:15" AND StartTime <= "16:15"
           OR EndTime >= "14:15" AND EndTime <= "16:15" )
     ) AS BookingsAtTime INNER JOIN FacilityForSport
       ON BookingsAtTime.FacilityID =
         FacilityForSport.FacilityID
WHERE Sport = "Basketball"
```

Refer nested solutions to team leaders for marking

Qu	Pt	Marking guidance	Total marks
4	1	**Mark is AO2 (analyse)** **B**; (Each product is only supplied by one supplier) **R.** if more than one lozenge shaded	1