



Rewarding Learning

ADVANCED
General Certificate of Education
2025

Software Systems Development

Unit A2 1:

Systems Approaches and Database
Concepts

[ADV11]

WEDNESDAY 21 MAY, AFTERNOON

**MARK
SCHEME**

General Marking Instructions

Introduction

Mark schemes are published to assist teachers and students in their preparation for examinations. Through the mark schemes teachers and students will be able to see what examiners are looking for in response to questions and exactly where the marks have been awarded. The publishing of the mark schemes may help to show that examiners are not concerned about finding out what a student does not know but rather with rewarding students for what they do know.

The Purpose of Mark Schemes

Examination papers are set and revised by teams of examiners and revisers appointed by the Council. The teams of examiners and revisers include experienced teachers who are familiar with the level and standards expected of students in schools and colleges.

The job of the examiners is to set the questions and the mark schemes; and the job of the revisers is to review the questions and mark schemes commenting on a large range of issues about which they must be satisfied before the question papers and mark schemes are finalised.

The questions and the mark schemes are developed in association with each other so that the issues of differentiation and positive achievement can be addressed right from the start. Mark schemes, therefore, are regarded as part of an integral process which begins with the setting of questions and ends with the marking of the examination.

The main purpose of the mark scheme is to provide a uniform basis for the marking process so that all the markers are following exactly the same instructions and making the same judgements in so far as this is possible. Before marking begins a standardising meeting is held where all the markers are briefed using the mark scheme and samples of the students' work in the form of scripts. Consideration is also given at this stage to any comments on the operational papers received from teachers and their organisations. During this meeting, and up to and including the end of the marking, there is provision for amendments to be made to the mark scheme. What is published represents this final form of the mark scheme.

It is important to recognise that in some cases there may well be other correct responses which are equally acceptable to those published: the mark scheme can only cover those responses which emerged in the examination. There may also be instances where certain judgements may have to be left to the experience of the examiner, for example, where there is no absolute correct response – all teachers will be familiar with making such judgements.

Fact finding techniques

AVAILABLE
MARKS

1 (a) Benefits:

Wide Reach: Potentially more representative sample/more feedback due to larger reach.

Cost-Effective: Questionnaires can be distributed to a large number of customers at a relatively low cost/cost effective compared to other fact finding techniques.

Time-Efficient: Respondents can complete questionnaires at their own convenience, and results can be collected quickly/quick electronic analysis/distribution.

Anonymity: Respondents may feel more comfortable providing honest answers due to the anonymity questionnaires offer.

Quantifiable Data: Questionnaires can generate quantifiable data (e.g. through rating scales or multiple-choice questions) that is easy to analyse.

Consistency: All respondents are asked the same set of questions, ensuring consistency in the data gathered.

[1] for each benefit (Max [2])

[2]

Drawbacks:

Limited Depth: Questionnaires often don't allow for in-depth responses, which can limit the quality of the insights gathered/limited depth due to lack of effort in response.

Low Response Rate: There is often a risk of a low response rate, especially if customers are not motivated to participate.

Misinterpretation of Questions/Answers: Without an interviewer to clarify, respondents may misinterpret questions, leading to inaccurate answers/no follow up questions.

Inflexibility: Once distributed, it's difficult to adjust questions if errors are found or if additional insights are needed/difficult to analyse poorly constructed questions.

[1] for any drawback (Max [2])

[2]

Complex analysis: difficult to analyse open-ended questions

(b) User Stories:

(i) Definition:

- Brief, simple/informal descriptions of a feature/function/requirement [1]
- From the perspective of the end-user/customer/stakeholder [1]
- Ensures that the development team builds software that meets a specific need [1]
- Articulates how a software feature will provide value to the customer [1]

[1] for each valid point (max [2]) [2]

Note: no credit for reference to multiple features. A user story only describes a single feature

(ii)

As a	receptionist
I want [1]	to easily select between various booking types [1]
So that [1]	multiple lessons can be booked easily [1]

[1] for 'I want'

[1] for 'So that'

[1] for each valid booking-related description of I want/so that [4]

10

2 Using the table below, identify which methodology each description refers to.

Description	Methodology
As the Project Ambassador, Mark will represent the interests of the employees at The Stables, acting as a liaison between them and the development team at Software Solutions.	DSDM
A Product Backlog is maintained for the proposed Event Booking portal and has a prioritised list of features that is open to changes based on the needs of The Stables.	SCRUM
To quickly develop an enhanced Lesson Booking system, Software Solutions engages stakeholders such as the receptionist in collaborative workshops (JAD sessions) to prototype and develop the system.	RAD
The software developers at Software Solutions are encouraged to be innovative and make bold decisions, like adopting new technologies for the Instructor Daily Schedule system, if it benefits the Mark and Diane's business.	XP
The requirements for The Stables' new Billing interface are well-defined, so a linear approach will be used for its development.	Waterfall

[1] per correct methodology. [5]

5

AVAILABLE MARKS

- 3 (a) • Ability to search for existing customers and populate their details/clear reference to permanent storage/retrieval of data.
- Form should be validated to ensure presence checks are in place for all required data/dates can only be booked at least one week in advance.
 - Only available dates and venues for selected lesson options should be displayed to avoid double bookings.
 - Prices should be accurately and automatically calculated/reference to billing.
 - Bookings should be clearly categorised to aid schedule creation for instructors.
- [1] credit any reasonable improvement related to Bookings (max [4]) [4]

(b) Answer should include some of the following points (not all points are needed for Level 3):

1. The same prototype is continuously refined
 - Developers keep building on the same model
 - It evolves gradually based on feedback, rather than restarting from scratch.
2. Users see working versions early
 - Users can try out incomplete versions of the system.
 - This helps them visualise and understand what is being developed.
3. Ongoing user feedback improves the system
 - Users regularly point out missing features or suggest changes
 - Developers can act quickly on this feedback.
4. Requirements become clearer through discussion and use
 - Users and developers work together to refine the systems requirements
 - Misunderstandings are cleaned up early.
5. Frequent testing leads to fewer problems later
 - Each version is tested, helping to catch and fix issues early.
 - This reduces the chances of major faults at the end.
6. The system evolves with changing needs
 - As users change their minds or discover new needs, the system adapts
 - This is especially helpful in fast-changing or uncertain environments
7. Better final outcome that suits users/leads to high quality system
 - The final system is much more likely to meet real-world needs
 - User satisfaction is higher, and less time is spent correcting issues after launch.

Sample Case study links:

End users such as Mark, Diane and the receptionist are keenly aware of the issues surrounding the booking process for the riding school, but none of them are technically minded enough to understand how these issues can be resolved in a computerised solution.

They need to be involved throughout each prototype iteration to assess its merits and drawbacks in order to inform the next iteration and also instil in them a greater understanding of their requirements.

For example, a prototype that displays the ability to capture customer information may prompt them to realise that they need to add functionality so that existing customers can be searched for, saving time and reducing duplication of work. Feedback like this will help the developers ensure that each subsequent prototype aligns more closely to the end user's needs and more closely to the final, high quality product.

Mark and Diane are conscious that bookings inform the schedule for instructors, and prototyping enables them to communicate to developers how they need to refine the way categories of the various lesson types are displayed, chosen and stored, so that the system can evolve to become a more efficient, high quality solution that enabled more accurate scheduling functionality.

Level 1 ([1]–[2])

Overall impression: Basic

The candidate provides a basic explanation of evolutionary prototyping, demonstrating limited understanding of its usefulness in software development methodologies. The response lacks depth and may only mention general points about prototyping without specific relevance to the booking system for the riding school. Connections to the scenario are minimal, and there is limited identification of how the end users and developers can work together within the context of prototyping to improve the system. The selection and use of appropriate form and style of writing are limited, with the organisation of material lacking clarity and coherence. There is little use of specialist vocabulary related to software development or prototyping. Presentation, spelling, punctuation, and grammar may be insufficient, leading to unclear communication of the intended meaning.

Level 2 ([3]–[4])

Overall impression: Good

The candidate offers a good explanation of evolutionary prototyping, showing reasonable insight into its role in software development methodologies.

The response addresses some benefits of evolutionary prototyping and the linkage to the booking process of the riding school is well-established, with the candidate making relevant points about how the prototyping process can help refine the system through communication between the users and the developers. There is a good selection and use of appropriate form and style of writing. The material is organised in a manner that provides some clarity and coherence. Specialist vocabulary related to prototyping and software development is used appropriately. Presentation, spelling, punctuation, and grammar are managed well enough that the intended meaning is generally clear.

AVAILABLE
MARKS

Level 3 ([5]–[6])

Overall impression: Excellent

The candidate delivers an excellent discussion on evolutionary prototyping, demonstrating a sound understanding of its value within software methodologies.

The response explains how iterative development and continuous feedback make evolutionary prototyping particularly useful for the riding school's booking system, providing specific examples of how the interaction between the users and the developers can help improve the functionality and quality of the solution, allowing it to evolve into the final system to the satisfaction of all stakeholders. The candidate demonstrates an excellent choice of form and style of writing, with the material being highly organised, clear, and coherent. There is excellent use of specialist vocabulary relevant to prototyping and software development.

Presentation, spelling, punctuation, and grammar are of a high standard, ensuring the intended meaning is conveyed effectively. [6]

10

4 (a) Answer should include some of the following points (not all points are needed for Level 3):

- Grace should set up regular communication channels (e.g. meetings, emails) with relevant stakeholders at all organisational levels of The Stables and Software Solutions:

Stables

- Owners (Mark, Diane)
- Administration staff (receptionist/other admin staff)
- Operational staff (instructors, grooms, stable hands)

Software Solutions

- Systems analyst
- Lead developer/Development team
- Testers

This will ensure that risks are identified early.

- Early communication enables proactive action. By identifying risks early, Grace can allocate resources or adjust timelines to minimise their impact on the project's success.
- Effective communication ensures that all stakeholders understand the risk response plan and their roles in mitigating the risks.
- Grace should maintain a risk log to document all identified risks, including their probability, potential impact, and mitigation strategies.
- The risk log should be regularly updated and shared with both The Stables' staff and the software development team to ensure transparency and collective awareness.
- Examples:
(Importance of communication)
 - The risk of double bookings of lessons could be raised by the receptionist or instructor early if there is open communication, allowing Grace to address the issue before it affects lesson scheduling.
 - By maintaining regular communication with the development team, they will have an opportunity to flag at an early stage the risk of delays in integrating the booking system with existing payment processing software. If integration takes longer than expected, it could delay entire project timeline, there regular communication is crucial.

(Risk Log)

- If there is a risk that the Billing and Payment module rollout could be delayed due to insufficient training for staff, the risk log should be monitored closely and adjustments made by Grace to the project timeline if necessary.

Note: Accept any relevant example that demonstrates how effective communication can help prevent or manage risks, or how risks can be recorded and monitored. Examples should be within in the context of The Stables and may address risks related to time, cost, scope, or other project factors.

Level 1 ([1]–[2])**Overall impression: Basic**

Candidate demonstrates a basic understanding of risk management and its connection to communication.

Limited reference to communication with stakeholders, addressing only The Stables or Software Solutions, but not both.

Limited explanation of how effective communication can help manage risks.

Limited examples provided. The candidate's selection and use of an appropriate form and style of writing are limited, with the organisation of the material lacking in clarity and coherence.

There is little use of specialist vocabulary. Presentation, spelling, punctuation, and grammar may be insufficient, leading to unclear communication of intended meaning.

Level 2 ([3]–[4])**Overall impression: Good**

Candidate demonstrates a reasonable understanding of risk management and its connection to communication.

There is some reference to communication with stakeholders from either The Stables and/or Software Solutions, though the explanation may lack depth or balance.

The explanation of how effective communication helps manage risks is clear, with some relevant points made, but may not be fully developed.

At least one relevant example is provided, showing an attempt to link communication with risk management in the context of the project.

The candidate's selection and use of an appropriate form and style of writing is adequate, and the material is generally organised with some clarity and coherence.

Some use of specialist vocabulary is evident, though not consistently accurate or thorough.

Presentation, spelling, punctuation, and grammar are generally sufficient to convey meaning, though minor errors may be present.

Level 3 ([5]–[6])**Overall impression: Excellent**

The candidate demonstrates a comprehensive understanding of risk management and its strong connection to effective communication. Clear and balanced reference is made to communication with stakeholders from both The Stables and Software Solutions, showing a thorough grasp of the need to manage communication across both areas. The explanation of how effective communication helps manage and mitigate risks is well-developed, detailed, and fully relevant to the project. Multiple relevant examples are provided, effectively linking communication to risk management in the context of the project and demonstrating deeper insight. The candidate's

selection and use of an appropriate form and style of writing is well-judged, with the material clearly and coherently organised. Specialist vocabulary is used accurately and effectively throughout. Presentation, spelling, punctuation, and grammar are consistently accurate, facilitating clear communication of the intended meaning. [6]

(b)

Statement	true/false
The life cycle for a development project like The Stables typically concludes with the planning phase. This ensures that execution and closure are well-guided by Grace.	False
As project manager, Grace will often rely solely on her intuition and experience rather than a feasibility study for project initiation.	False
In agile project management, Grace will use Gantt charts as the most effective tool for tracking progress, due to their flexibility.	False
Resource allocation is a significant focus for Grace during project planning.	True

[1] each correct true/false answer [4]

- 5 (a) Answer should include some of the following points (not all points are needed for Level 3):

Note: Design-stage testing (sometimes called walkthroughs or inspections) is about reviewing and dry-running diagrams, algorithms and requirements before any actual code exists.

Early Detection of Flaws:

- Testing at the design stage helps identify potential issues in the booking process before any code is written.
- Designing tests that simulate the booking process can reveal flaws in recording and retrieving bookings.
- Addressing these flaws before development begins improves the software's effectiveness
e.g. walking through a sequence diagram to verify that the client object is populated before the booking object is created, and messages between objects are accurate - could identify flaws in data flow before any code is written.

Improvement in Design Quality:

- The software can incorporate features to automatically validate data entry and flag issues.
- Testing these features with simulated scenarios replicating data entry at The Stables ensures robustness.
- Testing helps address data inconsistencies experienced by staff and customers
e.g. 1. reviewing class diagrams to determine if accurate relationships and cardinality are in place.
2. Creating algorithms (pseudo-code/flow charts) and manually tracing its execution on sample inputs via a dry run. That way, you catch off-by-one errors or infinite-loop risks before coding begins.

AVAILABLE MARKS

10

Risk Management:

- Early testing assesses risks related to data security, especially for billing and payment information.
- Identifying risks early allows for the design to include stronger security measures.

Facilitates Test-Driven Development (TDD):

- Using TDD at the design stage, as promoted by Extreme Programming (XP), ensures system components (e.g. booking and scheduling) function correctly from the start.

Saves Time and Cost:

- Catching design flaws early through testing avoids costly revisions later in development.
- This is particularly important given current inefficiencies in handling financial transactions and payroll at The Stables.

User Satisfaction and Reliability:

- Early testing ensures the system meets user expectations, ensuring reliability and ease of use.
- Addresses current frustrations with booking and equipment management, improving overall user satisfaction.

Level 1 ([1]–[2])**Overall impression: Basic**

The candidate provides a basic explanation of the significance of testing during the design stage, demonstrating limited understanding of the design and purpose of such software development practices.

The connection made to the case study is minimal.

The candidate's selection and use of an appropriate form and style of writing are limited, with the organisation of the material lacking in clarity and coherence.

There is little use of specialist vocabulary related to software testing.

Presentation, spelling, punctuation, and grammar may be insufficient, leading to unclear communication of intended meaning.

Level 2 ([3]–[4])**Overall impression: Good**

The candidate offers a good explanation of the importance of testing during the design stage, showing reasonable insight into the elements of an effective testing strategy.

The linkage to the case study is well-established.

There is a good selection and use of an appropriate form and style of writing. The material is organised in a manner that provides some clarity and coherence.

Specialist vocabulary related to software testing and development is used appropriately.

Presentation, spelling, punctuation, and grammar are managed well enough that the intended meaning is generally clear.

Level 3 ([5]–[6])**Overall impression: Excellent**

Explains in detail how testing during the design phase can uncover faults in the eventual system, and so saves time and money.

The candidate delivers an excellent discussion on the importance of testing during the design stage with a deep understanding of the principles of

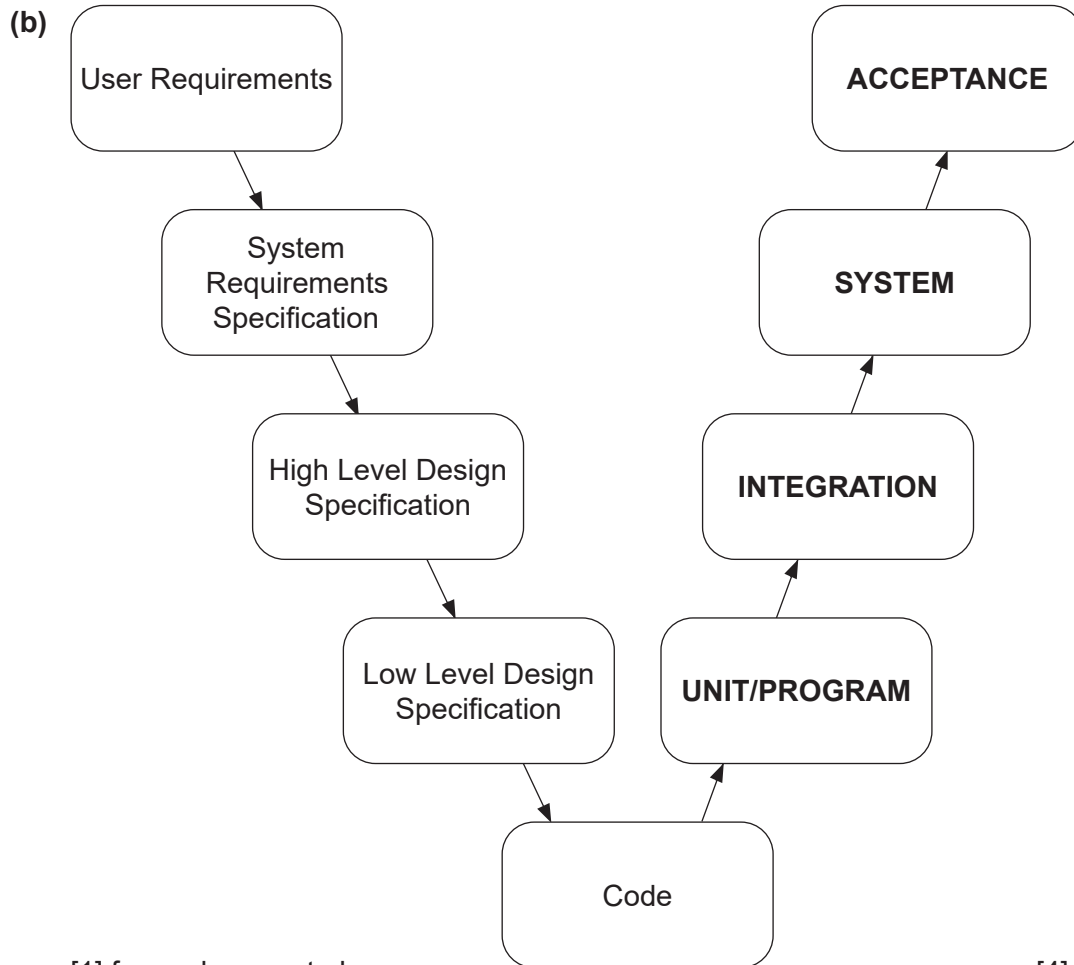
software testing and its relevance to the case study.

The analysis makes an excellent connection to the case study, with comprehensive suggestions for how testing could address specific issues identified in The Stables.

The candidate demonstrates an excellent choice of form and style of writing, with the material being highly organised, clear, and coherent.

There is excellent use of specialist vocabulary relevant to software testing and development.

Presentation, spelling, punctuation, and grammar are of a high standard, ensuring the intended meaning is conveyed effectively. [6]



[1] for each correct phase

[4]

10

6 (a) Answer should include some of the following points (not all points are needed for Level 3):

- A use case diagram is a visual representation that shows different users (actors) and how they interact with a system.
- For Diane, who is cautious about adopting new technology, a use case diagram can serve as a clear and straightforward tool to visualise the operations of the new computerised system and its benefits.
- The diagram can break down complex business processes into simple, understandable interactions between the user (Diane and her staff) and the system.
- It could show use cases like “Manage Bookings”, “Schedule Staff”, and

“Process Payments”, making it clear how these tasks are handled by the system.

- Diane, Mark, and other staff members, as well as customers would be identified as actors in the diagram.
- This helps Diane see who interacts with the system and in what capacity, reassuring her that the new system supports her current staffing structure.
- Each use case in the diagram can be linked to specific benefits such as time savings, error reduction, and improved customer satisfaction.
- The diagram can illustrate how different use cases are interrelated, showing Diane the integrated nature of the system.
- This demonstrates how changes in one area, like lesson cancellations, automatically update related areas, such as staff availability.
- By presenting a visual overview, the use case diagram provides a basis for discussion between Diane, the systems analyst at Software Solutions, and staff.
- It can help in addressing concerns and highlighting how new processes will be more efficient compared to the old ways.
- It can make it easier for her to appreciate how it addresses her current challenges and can lead to a more efficient and error-free operation.
- This should help reduce her reluctance to change by providing a clear depiction of the potential improvements.

Level 1 ([1]–[2])

Overall impression: Basic

The candidate provides a basic explanation of use case diagrams and how they might help Diane to better visualise the proposed system.

Linkage to the case study is minimal.

The candidate’s selection and use of an appropriate form and style of writing are limited, with the organisation of the material lacking in clarity and coherence.

There is little use of specialist vocabulary related to use case diagrams.

Presentation, spelling, punctuation, and grammar may be insufficient, leading to unclear communication of intended meaning.

Level 2 ([3]–[4])

Overall impression: Good

The candidate offers a good explanation of use case diagrams and how they might help Diane to better visualise the proposed system.

The linkage to the case study is well-established.

There is a good selection and use of an appropriate form and style of writing.

The material is organised in a manner that provides some clarity and coherence.

Specialist vocabulary related to use case diagrams is used appropriately.

Presentation, spelling, punctuation, and grammar are managed well enough that the intended meaning is generally clear.

Level 3 ([5]–[6])

Overall impression: Excellent

The candidate provides an excellent explanation of use case diagrams including suitable terminology and consistent reference to how they might help Diane to better visualise the proposed system.

The analysis makes an excellent connection to the case study, with examples of use cases, actors, and the relationships between them.

The candidate demonstrates an excellent choice of form and style of writing, with the material being highly organised, clear, and coherent.

There is excellent use of specialist vocabulary relevant to use case

diagrams.

Presentation, spelling, punctuation, and grammar are of a high standard, ensuring the intended meaning is conveyed effectively. [6]

(b) Reasons may include:

- Includes basic/inaccurate cardinality/relationships [1] Precise cardinalities are confirmed later (during design)/shows relationships are still being discovered and validated [1]
- Missing link table/entities [1] Link tables emerge later (via normalisation)/shows the diagram is a conceptual view, not yet fully normalised. [1]
- Missing non-atomic attributes [1] atomic decomposition is a design-time task/shows that the focus is capturing real-world concepts first, details later. [1]
- Missing data types [1] added later (in design) as not necessary for analysis phase [1]
- Missing methods [1] Behaviour is detailed later (in design)/analysis class diagrams often omit methods so that focus is on structure, not behaviour
- Missing visibility/access modifiers [1] added later as not necessary for analysis phase [1]
- Provides a high-level overview of the system [1] without delving into the implementation specifics. [1]
- Structured to aid communication [1] with non-technical stakeholders
- Used to understand requirements [1] example of requirement [1]

[2] points for each reason ([1] for point and [1] for how it aligns with analysis phase) [4]

Note: give credit for implication that missing features are yet to be added/implemented

10

7 (a) 2NF

CLIENT – Unchanged

CLIENT_HORSE

ClientID*, HorseID*

HORSE

HorseID, HorseName, HorseBreed, HorseDOB, HorseGender, StableNo, ArrivalDate, LiveryServiceID, LiveryType, ServiceStart, ServiceEnd, SpecialInstructions

CLIENT_HORSE_VETVISIT

ClientID*, HorseID*, VetVisitID*

VETVISIT

VetVisitID, VetID, VetFName, VetSName, VetClinicID, VetClinicName, VetVisitDate, VisitReason, VetFollowupDate, Notes

AVAILABLE
MARKS

CLIENT_HORSE_VETVISIT_TREATMENT

ClientID*, HorseID*, VetVisitID*, TreatmentID*

TREATMENT

TreatmentID, TreatmentDesc

[1] removal of horse details into new table

[1] removal of treatment details into new table

[1] for any correct new PK (VetVisitID/HorseID/TreatmentID)

[1] for any correct new FK (VetVisitID*/HorseID*/TreatmentID*)

[1] no other new table

[5]

(b) 3NF

CLIENT – Unchanged

CLIENT_HORSE - Unchanged

HORSE

HorseID, HorseName, HorseBreed, HorseDOB, HorseGender, StableNo, ArrivalDate, LiveryServiceID*, ServiceStart, ServiceEnd, SpecialInstructions

LIVERY_SERVICE

LiveryServiceID, LiveryType

CLIENT_HORSE_VETVISIT

ClientID*, HorseID*, VetVisitID*

VETVISIT

VetVisitID, VetID*, VetClinicID*, VetVisitDate, VisitReason, VetFollowupDate, Notes

VET

VetID, VetFName, VetSName

VET_CLINIC

VetClinicID, VetClinicName

CLIENT_HORSE_VETVISIT_TREATMENT - Unchanged

TREATMENT – Unchanged

[1] each for any **two** of:

- removal of only VetFName and VetSName details into new table (excluding PK)
- removal of only VetClinicName into new table (excluding PK)
- removal of only LiveryType into new table (accept removal of all livery service details)

[1] for removal of vetclinic details from HORSE_VETVIST table

[1] for any new PK (VetID/VetClinicID/LiveryServiceID)

[1] for any new FK (VetID*/VetClinicID*/LiveryServiceID*)

[1] for no change to any other table

[5]

AVAILABLE
MARKS

10

8 (a) Entity

- An object or concept (person, place, thing, or event) [1] about which data is to be recorded and maintained. [1]
- Each entity can be uniquely identified. [1]
- The entity is represented by a rectangle/labelled with a singular noun. [1]
- Has a set of attributes [1]
 - An entity typically corresponds to a table in a database [1]
 - Suitable example of entity and/or attributes [1]

Relationship

- A meaningful association between two entities [1]
- Relationships are labelled with verbs/example [1]
 - Implemented (in SQL) via foreign keys [1]

Cardinality

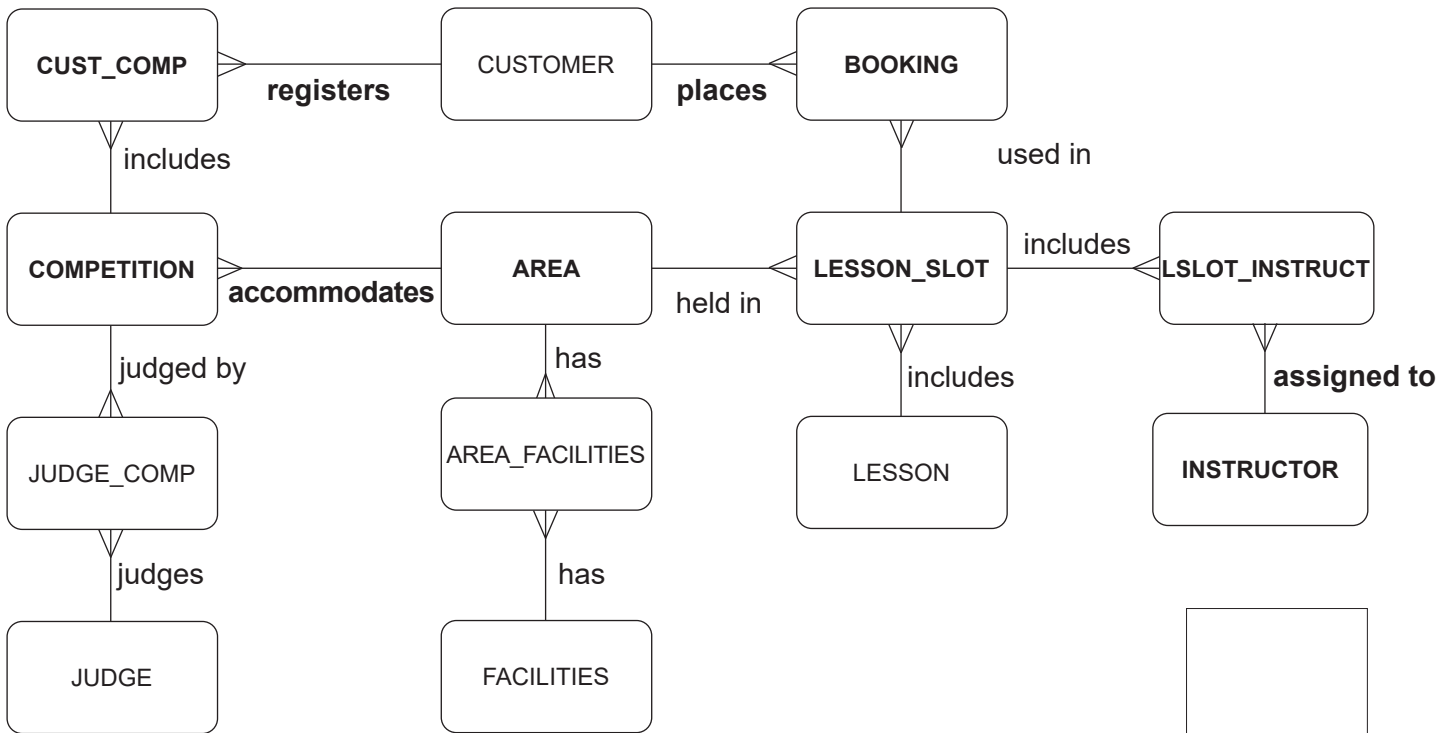
- Defines the multiplicity between two entities (i.e. how many instances of one can relate to instances of another) [1]
 - Expressed as 1:1, 1:N (one-to-many) or M:N (many-to-many)/ suitable example [1]

AVAILABLE MARKS

[1] for each valid point (Max [2] per term)

[6]

(b)



[1] for each correct answer

[11]

17

- 9 (a) (i) INSERT INTO LESSONSLLOT (DayOfWeek, StartTime, EndTime, LessonID, AreaID)

```

SELECT
    'Monday' AS DayOfWeek,
    '17:00:00' AS StartTime,
    DATEADD(MINUTE, L.Duration/Duration/30, '17:00:00') AS EndTime,
    L.LESSONID,
    1 AS AreaID,
FROM
    LESSON L
WHERE
    L.LessonID = 2

```

[1] each

[5]

- (ii) INSERT INTO LSLOT_INSTRUCT (LessonslotID, InstructorID)
VALUES
 (15, 18),
 (15, 27)

[1] insert into (Table)

[1] Each correct value row. Deduct [1] if VALUES keyword missing (or SELECT if used incorrectly)

Note: Accept two separate INSERT statements. Column names not necessary as long as values in correct order. Ignore quote marks on values

[3]

- (b) SELECT

```

    i.Forename AS InstructorForename,
    i.Surname AS InstructorSurname,
    COUNT(b.BookingID) AS TotalBookings,
    SUM(L.Duration) / 60.0 AS TotalDurationHours
FROM
    Instructor i
JOIN
    LSLOT_INSTRUCT li ON i.InstructorID = li.InstructorID
JOIN
    LESSONSLLOT s ON li.LessonslotID = s.LessonslotID
JOIN
    Bookings b ON s.LessonslotID = b.LessonslotID
JOIN
    Lesson L ON s.LessonID = L.LessonID

```

WHERE

b.BookingDate >= DATEADD(WEEK, 1, GETDATE()) AND

b.BookingDate < DATEADD(WEEK, 2, GETDATE())

GROUP BY

i.Surname,

i.Forename

ORDER BY

SUM(L.Duration) / 60.0 DESC

**if using DATEDIFF:*

WHERE

DATEDIFF(DAY, GETDATE(), b.BookingDate) >= 7 AND

DATEDIFF(DAY, GETDATE(), b.BookingDate) < 14

[1] SELECT includes instructor forename and surname

[1] Correct COUNT in SELECT (accept COUNT(*) or any field from the BOOKING table except HealthIssues)

[1] SUM(L.Duration) in SELECT

[1] Hours calculated in SUM (credit 60 although 60.0 ensures accuracy)

[1] FROM INSTRUCTOR/LSLOT_INSTRUCT/LESSONSLLOT/BOOKING/ LESSON

[1] All **five** tables joined correctly

[1] WHERE includes DATEADD or DATEDIFF

[1] Correct calculation in WHERE for week timespan (allow calculations for current week)

[1] Correct GROUP BY (2 fields required)

[1] Correct ORDER BY (credit SUM(L.Duration) without hour calculation) [10]

18

Note:

Use of aliases are not necessary if field exist in only one table e.g. Duration and BookingID.

Credit any acceptable alternative for a week timespan calculation.

Credit alternative variation for hours SUM(DATEDIFF(HOUR, StartTime, EndTime)) as BOD

Alias can be used for ORDER BY (do not double penalise if alias attached to incorrect calculation of hours)

Total

100