

THEME A: CONCEPTS · A3.2

Database design — Mark scheme

35 marks · covers A3.2.1–A3.2.7

How to mark. Each creditworthy point is on its own line and ends with a semicolon (;). A line beginning with ... is an *expansion* of the line above it, so the pair is worth the marks shown. A slash (/) separates acceptable alternative wordings. Words in (brackets) are not needed for the mark. Mark positively: credit what the student has got right rather than penalising what is missing, and accept any valid alternative not listed here. Do not award more than the maximum for a part. Where a later part uses a wrong answer from an earlier part correctly, award follow-through marks (FT). HL marks parts that assess higher level only content.

Q	ANSWER	MARKS
1(a) A3.2.1	Award [3] max Award [1] per point up to [3] max. Both schemas must be referred to. The conceptual schema is a high-level, abstract description of what data the system holds and how the entities relate, independent of any particular database system; It is concerned with entities such as members, bikes and hires, not with how they are stored; The physical schema describes how the data is actually stored on the storage device; It covers file organisation, indexes, partitioning and access paths; ...(so) the conceptual schema can stay the same even if the physical storage is changed to improve performance;	3
1(b) A3.2.3	Award [2] max Award [1] for the point and [1] for the expansion. Any one problem. If the start time were stored as text rather than a date/time type, the database could not compare or sort it correctly; ...(so) a query for all hires in a date range would return the wrong rows, or none at all; Date arithmetic would not be possible on a text field; ...(so) the duration of a hire could not be calculated by the database; A text field accepts any format; ...(so) inconsistent entries such as 02/04/2026 and 2026-04-02 could both be stored, breaking data consistency;	2
2(a) A3.2.2	Award [1] max The number of instances of one entity that can be associated with an instance of the other entity, for example one-to-one, one-to-many or many-to-many;	1
2(b) A3.2.2	Award [4] max Award [1] per point up to [4] max. Three entities drawn and labelled MEMBER, BIKE and HIRE; A relationship drawn between MEMBER and HIRE; A relationship drawn between BIKE and HIRE; Cardinality shown correctly as one-to-many from MEMBER to HIRE (one member makes many hires); Cardinality shown correctly as one-to-many from BIKE to HIRE (one bike is used for many hires);	4

Q	ANSWER	MARKS
	HIRE resolves the many-to-many relationship between MEMBER and BIKE. Accept crow's foot, Chen or UML-style notation provided the cardinality is unambiguous. Also accept a direct many-to-many between MEMBER and BIKE with HIRE shown as the linking entity. Do not award the cardinality marks if the diagram shows only unlabelled lines.	
2(c) A3.2.2	Award [1] max Optional (on the HIRE side) / modality of zero; Accept any wording making clear that a MEMBER need not participate in the relationship.	1
3(a) A3.2.5	Award [2] max Award [1] per point up to [2] max. Every field holds a single, indivisible value; There are no repeating groups and no lists of values held in one field; ...(so) a field could not hold two bike IDs separated by a comma;	2
3(b) A3.2.5	Award [4] max Award [1] per point up to [4] max. Both forms must be addressed and the type of dependency named. 2NF requires the table to already be in 1NF and to have no partial-key dependencies; A partial-key dependency is where a non-key attribute depends on only part of a composite primary key, rather than the whole of it; So 2NF only becomes an issue when the primary key is composite; 3NF requires the table to already be in 2NF and to have no transitive dependencies; A transitive dependency is where a non-key attribute depends on another non-key attribute rather than directly on the primary key; ...(so) 2NF removes dependencies on part of the key, and 3NF removes dependencies on something that is not the key at all;	4
4(a) A3.2.4	Award [1] max (BikeID, StartTime); Neither attribute is unique on its own in the data: B07 appears three times, and 2026-04-03 09:05 appears twice because two different bikes were hired at that moment. Only the pair identifies a row. Do not accept StartTime alone, and do not accept MemberID as part of the key, since a member may hold several hires. The key follows from the stated rule that a bike can begin only one hire at a given start time; award [1] to a candidate who instead proposes (MemberID, StartTime) <i>and</i> states the assumption that a member cannot hire two bikes at once.	1
4(b) A3.2.5	Award [2] max Award [1] for identifying the dependency and [1] for explaining why it is partial. BikeModel depends on BikeID; ...(and) BikeID is only part of the composite key (BikeID, StartTime), so BikeModel depends on part of the key rather than the whole key; Do not credit MemberName or MemberEmail here: MemberID is not part of the key, so those are transitive dependencies, not partial ones.	2
4(c) A3.2.6	Award [5] max Award [1] for the MEMBER table with the correct primary key; [1] for the BIKE table with the correct primary key; [1] for the HIRE table with the correct composite primary key; [1] for MemberID as a foreign key in HIRE; [1] for no remaining partial or transitive dependencies (DurationMins left in HIRE, nothing duplicated).	5

Q	ANSWER	MARKS
	MEMBER (<u>MemberID</u>, MemberName, MemberEmail) BIKE (<u>BikeID</u>, BikeModel) HIRE (<u>BikeID</u>, <u>StartTime</u>, MemberID, DurationMins) BikeID in HIRE is both part of the composite key and a foreign key to BIKE. Accept an equivalent design that introduces a surrogate HireID as the primary key, provided (BikeID, StartTime) is then given a uniqueness constraint and both foreign keys are present. Accept tables drawn out in full instead of the bracket notation.	
5(a) A3.2.4	Award [3] max Award [1] per point up to [3] max. A primary key guarantees every row can be identified uniquely, so duplicate records cannot be created; A foreign key enforces referential integrity, so a hire cannot refer to a member or a bike that does not exist; The database will reject an attempt to delete a member who still has hires, or will cascade the change consistently; Each attribute has a defined data type and constraints, so invalid values are refused on entry; Because each fact is stored in exactly one table, there are no conflicting copies to fall out of step;	3
5(b) A3.2.4	Award [2] max Award up to [1] per difference, [2] max. A composite key is a primary key made up of two or more attributes together, whereas a foreign key is an attribute that refers to the primary key of another table; A composite key identifies rows within its own table, whereas a foreign key creates a link between two tables; Accept the observation that one attribute can be both, as BikeID is in the HIRE table.	2
6 A3.2.7	Award [5] max This is an evaluate question (AO3), so both sides are required and applied to the dashboard. Award [1] per point up to [5] max, but award a maximum of [3] if only one side is given, and a maximum of [4] without a supported judgement. Arguments for denormalizing The dashboard is read-intensive: it queries constantly but never changes the data it displays; Removing the joins means the query reads from one table, which is markedly faster; A refresh every few seconds multiplies that saving, reducing load on the server; The query itself becomes simpler to write and to maintain; Arguments against denormalizing The same data is now stored in more than one place, reintroducing redundancy; Every update must be applied everywhere the data is duplicated, or the copies disagree; ...(so) the dashboard could show a station as having bikes that have already been taken; More storage is used, and the write path becomes slower and more complex; The integrity guarantees that normalization provided are now the application's responsibility rather than the database's; Judgement (credit any supported conclusion, for example) Denormalizing is justified here because the dashboard is read-heavy and a few seconds of staleness is acceptable for a bike count, provided the normalized tables remain the authoritative copy and the duplicated data is refreshed automatically;	5

Q	ANSWER	MARKS
	...or: a cached query result or a materialized view would gain most of the speed without permanently duplicating the data;	

Total for this paper: 35 marks