

AQA Computer Science A-Level

**4.10.3 Database design and
normalisation techniques**

Past Paper Questions

Additional Specimen AS Paper 1
June 2012 Comp 3

9	(b)	<p>Data is atomic // no repeating groups (of attributes); R No repeated columns/attributes/data/values</p> <p>No partial (key) dependencies // No (non-key) attribute depends on part of the primary key but not the whole of it // all non-prime attributes are (functionally) dependent on the whole of every candidate key // (non-key) attributes depend on the whole key;</p> <p>No non-key dependencies // No transitive dependencies // (non-key) attributes depend on nothing but the key;</p> <p>Every (non-key) attribute is dependent upon the key;</p> <p>Every determinant is a candidate key;</p> <p>A "field" for "attribute" A "part" for "partial"</p> <p>MAX 2</p>	2
----------	-----	--	---

9	(f)	<p>ONE MARK FOR PRINCIPLE AND MAX TWO MARKS FOR IMPLEMENTATION</p> <p>Principle:</p> <p>Create a new table (A link table) (BookCopy); through which Book and Loan tables will be (indirectly) linked;</p> <p>Implementation details using a new primary key:</p> <p>Create a new <u>unique ID/key field</u> (e.g. CopyID) (for each copy); Store the BookID and the CopyID in the new table; Replace the BookID in the Loans table with this CopyID;</p> <p>Note: In this implementation, CopyID is unique, i.e. BookID 1 and 2 cannot both have CopyID 1.</p> <p>Implementation details using a composite key:</p> <p>Create a new field CopyID; Composite key formed by BookID and CopyID; TO if composite key is clearly in book table or loan table Store the BookID and the CopyID in the new table; R adding CopyID to Book table as this would created data redundancy but this does not talk out other mark scheme points Add the CopyID field to Loans table; R replace BookID with CopyID</p> <p>Note: In this implementation, CopyID is not unique, e.g. BookID 1 and 2 can both have CopyID 1.</p>	
---	-----	---	--

June 2017 Paper 2

10	2	<p>1 mark for AO2 (analyse) and 1 mark for AO1 (understanding)</p> <p>AO2 (analyse) – 1 mark: A person may own more than one car // a person may bring different cars to the garage; It might be desired to store details of an owner when the car they own is not yet known; A. A car might be owned by more than one person (at different times) A. Easier to transfer car from one owner to another</p> <p>AO1 (understanding) – 1 mark: Avoid storing owner details once for each car they own / multiple times; Avoid having to input owner details once for each car they own; To transfer car between owners would only have to change one attribute in the car relation; Minimise data duplication // no unnecessary repeated data; A. Reduce for minimise</p>	2
		<p>Eliminate data redundancy; A. Reduce/minimise for eliminate Eliminate data inconsistency // improve consistency // avoid inconsistency problems; Eliminate update anomalies; A. Example in context Eliminate insertion anomalies; A. Example in context NE. Fewer errors when updating/inserting/deleting without concrete example or good explanation NE. Saving space/memory NE. Easier to query</p>	

10	7	<p>All marks AO2 (analyse)</p> <p>1 mark: Create a new relation to identify which make/model(s) of car each part can be fitted to;</p> <p>A. Use of a relation name that clearly identifies the purpose eg <code>PartToFitMakeModel</code> instead of an explanation</p> <p>A. If it is just stated that a new relation is creation if the attributes in the relation make its purpose clear</p> <p>NE. A relation to link the <code>Part</code> and <code>Car</code> relations</p> <p>2 marks from: Store the attributes <code>PartID</code>, <code>Make</code> and <code>Model</code> in the new relation;</p> <p>I. Inclusion of additional attributes Make the <code>PartID</code>, <code>Make</code> and <code>Model</code> / all the attributes the entity identifier;</p> <p>A. The creation of a new field as an entity identifier for this relation if it is explained that a constraint would also need to be added to ensure that it is not possible to record twice in the relation that a particular part could be fitted to a particular make and model of car</p> <p>Accept answers by example, such as: <code>PartToFitMakeModel (PartID, Make, Model)</code></p> <p><u>Alternative Response</u></p> <p>1 mark: Create two new relations, one to associate an entity identifier with each make and model of car (eg <code>MakeModelID</code>) and one to link the parts to this new relation.</p> <p>A. If it is just stated that new relations will be created if the attributes in the relations make their purpose clear</p> <p>2 marks from: Store the attributes <code>Make</code> and <code>Model</code> with a new entity identifier (eg <code>MakeModelID</code>) in one of the new relations;</p> <p>Store the <code>PartID</code> in the other new relation together with the entity identifier from the first new relation (eg <code>MakeModelID</code>);</p> <p>Make the <code>PartID</code> and <code>MakeModelID</code> the entity identified in the second new relation;</p> <p>A. The creation of a new field as an entity identifier for this relation if it is explained that a constraint would also need to be added to ensure that it is not possible to record twice in the relation that a particular part could be fitted to a particular make and model of car</p> <p>Accept answers by example, such as: <code>UniqueMakeModel (MakeModelID, Make, Model)</code> and <code>PartToFitMakeModel (PartID, MakeModelID)</code></p> <p>A. Table or entity for relation. A. Field for attribute. A. Primary key for Entity Identifier.</p>	3
----	---	--	---

June 2013 Comp 3

9	(a)	<p>What means: every attribute (in relation) is dependent on the key; the whole key; and nothing but the key; R. Everything OR (relations) contain no repeating groups (of attributes) // data is atomic; no partial dependencies; no non-key dependencies; R No repeated columns/attributes/data OR every determinant (in the relation) is a candidate key;;</p>	4
----------	------------	---	----------

		<p>MAX 2 Why important: Eliminate update anomalies; A Example Eliminate insertion anomalies; A Example Eliminate deletion anomalies; A Example Eliminate data inconsistency // improve consistency // avoid inconsistency problems; *Minimise data duplication // no unnecessary repeated data; A Reduce for minimise R eliminate *Eliminate data redundancy; A Reduce/minimise for eliminate NE Easier to update/insert/delete without concrete example or good explanation NE Less errors when updating/inserting/deleting without concrete example or good explanation NE Saving space/memory NE Easier to query Award marks to points made anywhere across 9(a) Can only award one of the two marks indicates by asterisks (*) MAX 2</p>	
--	--	---	--

Specimen Paper 2

09	2	<p>All marks AO1 (knowledge)</p> <p>Any 2 from: Data is atomic // no repeating groups (of attributes); R. No repeated columns/attributes/data/values No partial (key) dependencies // No (non-key) attribute depends on part of the primary key but not the whole of it // all non-prime attributes are (functionally) dependent on the whole of every candidate key // (non-key) attributes depend on the whole key; No non-key dependencies // No transitive dependencies // (non-key) attributes depend on nothing but the key; Every (non-key) attribute is dependent upon the key; Every determinant is a candidate key; A. 'field' for 'attribute' A. 'part' for 'partial' MAX 2</p>	2
-----------	----------	--	----------