
2910209 Database systems

Examiner's report: Zone B

Comments on specific questions

Question 1

The first part of this question required candidates to draw an Entity/Relationship diagram based on the description of a scuba-diving holiday company. Some candidates had difficulty dealing with the most challenging part of this description, which required them to see that both Holiday-Groups and individual Holiday-Makers were entity types in their own rights. For example, cabins were occupied by individual Holiday-makers, not by Groups. Also, some candidates did not distinguish between a Dive-event, which took place at a single Location but had a limited duration, and the Location itself.

The second part required candidates to frame normalised relations which could embody the entities, attributes and relationships discovered in the first part. Here, many candidates were less sure of themselves. Some ingenuity had to be exercised in designing some of the relations, since composite key values for some of the entity types were not obvious, and could in fact be designed in more than one way. Nonetheless, most candidates managed to capture most or all of the required information in a set of relations.

One hint which may prove helpful to candidates tackling an E/R Diagram question is this: the entity types which usually give us problems are the abstract ones: often things which combine two or more concrete entity types in some sort of relationship. A good way to uncover these is to construct a set of relations embodying the information upon which the E/R diagram is to be based, first. Then work backwards, under the assumption that each relation is an entity type.

Question 2

Candidates who attempted this question needed to understand the concept of functional dependency, the anomalies that can happen when relations are not fully normalised, and what to do in order to normalise unnormalised relations. They also needed to know the definitions of a number of terms used in relational theory.

Most candidates were successful in answering the first, third, and fourth parts of the question, but when tackling the second part, many of them revealed some serious misunderstandings of just what Update, Deletion, and Insertion Anomalies are. These terms have specific meanings. Some candidates evidently believed that any error involving an update was an example of an Update Anomaly, and similarly for Insertions and Deletions.

Candidates who encounter a similar question in future might find it helpful to do the following: if they think they have provided an example of a normalisation anomaly, they should check to see if this example would

also be possible with a relation in third normal form. If it would, it is not an example of normalisation anomaly, but of some other kind of error.

Question 3

This question tested candidates' mastery of SQL, and most candidates did not find it challenging. A very few candidates confused 'GROUP BY' with 'ORDER BY', and 'HAVING' with 'WHERE'. Candidates preparing for next year's examination, when studying SQL, should be sure to have sight of the exams for the last two or three years, and not just this one, where the questions were rather narrowly focussed.

Question 4

All parts of this question aimed to test candidates' knowledge of concurrent processes, as applied to databases. Almost all candidates did well on the part A of the question, which required them to describe one sort of problem, of their choice, which could arise when more than one user had update access to a database at the same time, and then to describe a solution. A few answers were rather vague, and gave evidence of recent hasty memorisation attempts rather than of deeper understanding.

Part B was a standard question which required candidates to describe five different scenarios for recovery from a crash, with updates that started before, or after, a checkpoint, and which finished before, or after it, or which were running when the crash occurred. This textbook example was answered well by almost every candidate who attempted it. Part C essentially described the 'deadly embrace' problem, without naming it, and required candidates to outline a solution. Most candidates could describe the problem, but a few had difficulty describing (as opposed to naming) one of the standard approaches to dealing with it. This question, in one form or another, continues to appear in database exams, and candidates preparing for the next exam should make sure they can explain how the 'deadly embrace' can be dealt with.

Question 5

The few candidates who opted for this question found it easy in parts, but often foundered on one or two sections. Some candidates evidently chose this question in desperation, not having prepared for the rather standard questions on the rest of the exam. Thus there were some wild guesses about 'vertical fragmentation', and the odd silly attempt to base a definition of 'data dictionary' from definitions of the words themselves taken in isolation. ('Dictionary of data' by itself will not do, unfortunately.) The last part of this question was the most difficult for many candidates, who struggled to present a clear and obvious example of a query that could be answered in more than one way, with one of the ways being much more efficient than the other. Candidates preparing for the examination next year should be sure that they can show how a join actually works, because understanding this – and in particular, understanding that it can potentially engage a large amount of processor time – is key to understanding how an optimiser could improve things, by executing sub-queries that greatly reduce the relations to be joined. Concrete examples, as usual, are the candidate's friend.