Examiners' commentaries 2017–2018

CO2209 Database Systems – Zone A

General remarks

This year's examination did not deviate, in terms of results, from those of recent years, although candidates did have the challenge of dealing with a paper which was wordier than usual. The great majority of candidates passed, and some did very well. However, there were a few failures, some of which were so complete that it is hard to believe that the candidates who did so badly had done any revision at all.

Comments on specific questions

Question 1

This question described a textile processing operation with many branches spread out around a large geographical area, and a central warehouse which distributed imported textiles of various sorts to the regional branches. This involved quantities of textiles being removed from one location and, after a period in transit, being stored in another. This provided a foundation for asking questions about concurrency, and the meaning of a 'transaction' as the term is used in the database world. Many candidates who could successfully answer the book-work question 'what is a transaction' proved unable to recognize one when presented with a sequence of SQL statements, where some sub-sequences were 'logical units of work' – that is, they all had to be completed in order for the database to remain consistent – and others stood alone. Candidates who ran aground here fell back on the 'normal' meaning of a transaction as just an operation on a database. There is always a difficulty in going from the abstract to the concrete and this is an area where that is especially true, which provides a marker for next year's coursework setter.

Question 2

This question presented candidates with a classic database problem: parts being manufactured, then assembled into a larger component, with these being distributed to re-sellers. It was necessary to draw up an Entity/Relationship Diagram which captured the entity types and relationships among them, and then to draw up a normalized relational schema. A final part of the question posed an extension of the scenario, which proved more problematic for many. The first part of the question was almost universally completed in a satisfactory way, but the second part revealed some weaknesses among a significant number of candidates. Those who had trouble here had difficulty in identifying the Primary Keys of the relations, and/or in creating normalized relations. Those who found the second part challenging generally did poorly on the last part: this was an example of the 'Connection Trap', where spurious tuples are generated upon re-joining two binary relations improperly derived from a ternary one.

Question 3

This question consisted of several sub-questions, each looking at some aspect of relational design. The first section showed a relation and offered several choices of combinations of attributes as its primary key. Most candidates did well here, but the number who did not was unsatisfactorily high. Understanding functional determinancy and key choice is at the heart of relational design, and getting this right is critical. The second section gave an example where it was important to recognize that normalization is not always desirable. Most candidates grasped this. The third section invited candidates to consider solving the problem of choosing a primary key by simply giving a unique number to each tuple. Far too many did not see that doing this obviates the whole purpose of having a designated primary key, namely, to prevent inherently impossible tuples from being added. The next section required a short essay on indexing, to which almost all candidates were able to respond well, although a few candidates did not realize that indexes have to be updated when a relation is updated and that this can affect performance. Following the indexing question was a question asking whether a read-only database would require concurrency control. Most candidates understood that it would not, although a few confused concurrency control with the problem of controlling access to a website by too many users at the same time. This section also asked about 'physical data independence', which almost all candidates were able to answer However, many answers gave the impression of being simply regurgitated definitions, without understanding that 'physical data independence' does not mean total 'independence' of the physical data (whatever that might mean) but independence, at the user level, from the way the data is physically organized and stored. The final part of this question asked for some definitions of basic terms, and was well answered. One persistent minor irritation was the use of a word in its own definition. Very few candidates could not answer these, which indicated that they had done no revision at all.

Question 4

This was a classic 'functional dependency' and 'normalization' question, a staple of every database examination that has ever been given anywhere in the world. Although most candidates could get most of the marks on this question, it was clear that some candidates believe that 'anomaly' is a synonym for 'error'. It is not. In this context, an 'anomaly' is a possible mistake that could be prevented via normalization. Normalization by itself cannot, sadly, prevent all errors from happening.

Some poor responses here were:

- claiming that an unnormalized relation would necessarily have update anomalies – not true, it simply makes them possible
- assuming that deletion of a tuple in an unnormalized relation always results in loss of information we want to keep – not true, it simply makes it possible.

Candidates reading this for guidance on future examinations should make very sure that they understand just what an 'anomaly' in the database context is.

The last part of this question presented a relation that met the Boyce-Codd criteria, yet was a poor design, and invited candidates to propose a better design. Most did, but there enough who did not to provide future coursework setters with one more issue which needs to be addressed when candidates do their practical work.

Question 5

This was 'the SQL question'. Or rather, the 'SQL question' and the 'NoSQL question'. The first part presented several relations, and then asked candidates to frame SQL statements that could extract information from them. This was not overly difficult for most candidates and most questions, but the usual SQL confusions appeared: not understanding the difference between GROUP BY and ORDER BY; not understanding how to frame a 'negative' question that requires a set difference operation; not understanding the unique qualities of 'NULL'. The last three sections of this question were more 'practical' in nature: how to speed up a sluggish database; how to test the validity of a query; when a relational database might be a poor fit to a given problem. Answers here were generally satisfactory, with some of the outstanding ones reflecting the fact that some candidates actually have many years of practical experience maintaining live databases.