Examiners' commentaries 2015–16

CO2209 Database systems – Zone A

General remarks

This year, the coursework assignment included reading the previous two years' *Examiners' commentaries* for Database systems. Reports for both zones summarise the most frequent mistakes made by candidates taking these examinations. As similar mistakes tend to arise year after year, it was hoped that focusing attention on them would help improve student performance in the examination.

This year – as last year – some candidates appear to have benefitted from the VLE discussion board. This platform enables students to post proposed solutions to questions from past examination papers. These elicited responses from both the VLE tutor and other students; and, in some cases, they provoked extensive discussion. There is some evidence that this facility – together with the coursework assignment focus on past errors – was helpful to many, including those who did not directly take part in the discussions. We hope this practice will grow in future years.

Comments on specific questions

Question 1

The first section of Question 1 presented candidates with a scenario involving a sewing machine factory assembling different models of sewing machines from various components and its network of distributors to whom it shipped the assembled models. Candidates were required to draw an Entity/Relationship (E/R) diagram to model these relationships. Few candidates had serious difficulty with this part of the question: the fact that the database world does not have a single standard for drawing such a diagram always requires examiners to show latitude in interpreting student efforts. A few candidates omitted to show all cardinality and/or participation constraints. Sometimes the discernment of such elements requires close reading of the question.

Candidates were then required to derive a normalised relational schema from the E/R diagram. Again, most candidates were able to do this with little difficulty; although a few neglected to indicate the Primary Keys of their relations. Following the diagram, this schema merely recorded what particular components were used in which sewing machine models and which models were delivered to which shops.

The next part of this question required the relational schema to be modified to show the projected dates for component deliveries; and the quantities shipped on those dates. The key idea here was that this required not just the addition of dates and quantity columns; but also a change of Primary Key in the amended relation to include the date of shipment. (A guiding rule of thumb for those studying relational design is that dates and times are often part of the Primary Key as they often distinguish otherwise-identical tuples from each other.) This task proved particularly challenging for some candidates.

Candidates were then required to consider what relational schema would capture relationships in circumstances where each component type would be supplied by only one supplier. Some candidates did not discern that this state of affairs would allow one, in a normalised schema, to have one relation recording which supplier supplied which component; and another showing dates and quantities for each component. Not all candidates recognised this.

However, overall this question was well-answered, and the problems some candidates had in identifying Primary Keys had been common in the past.

Question 2

Question 2 dealt with the problem of normalising an existing table given in un-normalised form. Candidates were required to identify its Primary Key; find its Functional Dependencies; comment on the insertion, deletion and modification anomalies in it; and then bring it to Boyce-Codd Normal Form. It is to candidates' credit that they noted, but were not thrown off track by a mistake in the wording of the question where the word 'update' appeared instead of the word 'insertion'.

The only notable difficulties here were those often encountered in discussing the issue of anomalies; namely, that not every possible error is an example of an anomaly that can be prevented by normalisation; and that not all deletion of data is deletion of unwanted data. If candidates encounter this type of question, they should always ask themselves, when writing on examples of a particular type of anomaly: (1) 'would normalisation prevent this?; and (2) 'if a particular tuple were deleted, is there any **unwanted** deletion of particular attributes taking place?'

Finally, candidates needed to consider a situation in which performance issues might militate against full normalisation. The example given was a relation holding street addresses, postal codes and cities, where postal code functionally determined city. Normalisation involves splitting up relations – in this case, creating two tables, one holding street addresses and their corresponding postal codes and the other holding postal codes and their corresponding cities – implying that their re-join will be necessary to answer certain queries; or to provide certain information. Joins are among the most costly operations (in terms of performance) in database systems; so, if normalisation means the regular re-join of large relations, it may be contra-indicated. In addition, some dependencies between data items are more tightly coupled than others – in this case, post-codes and cities – and, if the coupled pair are carefully checked for correctness upon initial tuple insertion, then the only price paid for not normalizing is extra storage, which today is not the problem that it was when relational databases were first introduced.

Questions 3 and 4

Whereas the first two questions required candidates to apply principles of relational design to concrete examples, the third and fourth questions mainly involved writing answers to questions about topics from the database syllabus. Few candidates chose Question 3—those who did clearly did so because they had revised the (fairly standard) topics it examined. One or two candidates showed some confusion about 'horizontal fragmentation'; it might be helpful if students studying this topic simply thought of it as turning one relation into two with the same schema but with the two relations distinguished by the frequency with which they are accessed. Accessing a large relation with many 'dead' tuples is likely to waste processing time. An example of the same principle is found among people who answer 'directory assistance' inquiries on the telephone system. As a significant proportion of such inquiries are about a small number of places (airports, bus stations, taxi services, hospitals, etc.), the operators keep a sheet of paper displaying the phone numbers of these frequently-required places where they can easily see it. The operators have 'horizontally fragmented' their database of phone numbers.

Although Question 4 was also a 'short essay' style question and was chosen by many candidates, it combined routine questions, worth 14 marks—requiring candidates to know basic definitions used in the relational model—with questions that required a measure of original thinking. One of these asked candidates how SQL differed from traditional programming languages like Java or Python. Most candidates who attempted this simply listed some features of SQL, instead of noting that SQL is a declarative – and not a procedural – language. Two other questions concerned Primary Keys: could we have a relation without one; and why do we have them in the first place? Unfortunately, only a few candidates who attempted this wrote good answers to this question. Therefore, candidates preparing for next year's examination are advised to prepare themselves for similar questions – that require more than simple replication of notes but rather deeper speculation about the reasons we do things in a certain way in database design. Use of the interactive forum to discuss these issues would be a beneficial way to prepare for such questions.

Question 5

This was the 'SQL question', attempted by nearly all candidates, and only a very few candidates found major difficulties with it. The types of problems that appeared here appear every year. Candidates were required to frame SQL queries: the errors included using = NULL instead of IS NULL, confusing ORDER BY with GROUP BY and equating WHERE with HAVING. Candidates preparing for next year's examination are strongly advised to be absolutely clear about the circumstances in which GROUP BY and HAVING are used. Another area where many students run into problems are queries which require using the set difference operator. Such queries are often found where a 'negative' is being used, such as finding all doctors who have not qualified in rhenology. Such queries may superficially appear to require the 'not equals' operator; but this is deceptive where the relationship between the data being retrieved and the attribute being referenced is one-to-many.

Summary/Conclusion

Candidates reading this report and looking at past examination papers will note that there are few 'surprises' in the examinations. If you work though the course material as outlined in the **Database systems** subject guide, complete the coursework assignments, look over past examination papers and engage with the online forum, you will find that the examination you sit is one in which you are able to do well.

→

Examiners' commentaries 2015–16

CO2209 Database systems – Zone B

General remarks

This year, the coursework assignment included reading the previous two years' *Examiners' commentaries* for Database systems. Reports for both zones summarise the most frequent mistakes made by candidates taking these examinations. As similar mistakes tend to arise year after year, it was hoped that focusing attention on them would help improve student performance in the examination.

This year – as last year – some candidates appear to have benefitted from the VLE discussion board. This platform enables students to post proposed solutions to questions from past examination papers. These elicited responses from both the VLE tutor and other students; and, in some cases, they provoked extensive discussion. There is some evidence that this facility – together with the coursework assignment focus on past errors – was helpful to many, including those who did not directly take part in the discussions. We hope this practice will grow in future years.

Comments on specific questions

Question 1

The first section of Question 1 presented candidates with a description of a property-management company whose employees looked after owners' properties rented out to tenants. Candidates were required to draw up an Entity/Relationship (E/R) diagram showing how these 'entity types' were related to each other, showing both cardinality and participation constraints. Almost all candidates presented acceptable solutions to this part of the question, although, here and there, careless reading of the initial conditions resulted in the loss of marks. (In database design, we **must** pay special attention to things that in ordinary life we are not so meticulous about, such as whether one thing must be related to another, and whether or not there is a maximum number of related objects. For example, under the rubric 'favourite movies', you may choose 'none', 'one', or 'many', but you must have a minimum, and maximum, of two parents (although recent advances in reproductive medicine have begun to alter this)!

Following the drawing of the E/R diagram, candidates were required to draw up a relational schema—basically, instructions for creating relations—which could incorporate the relationships shown in the diagram. This proved particularly challenging for some candidates, as has been the case in previous years. As a property can have (according to the description) only one Owner, one Tenant, and one managing Branch Office, all of these relationships can be held in a single relation, with the Property's PostalAddress as the Primary Key.

A hint: candidates who encounter an 'Entity-Relationship' question that requires drawing an E/R diagram, and then showing the corresponding relational schema, may find it helpful to work out the schema first, and then draw the diagram. Relational schemas are one level less abstract than

E/R diagrams and, for this reason, it is easier to rationalise about them.

Whether you choose to tackle the E/R diagram and then the relational schema, or vice versa, it is important to check that they correspond.

Question 2

Question 2 showed candidates an unnormalised table containing information about plots of land. They had to name its Primary Key, identify the Functional Dependencies among the table's attributes, give examples of update, deletion and insertion anomalies that can occur if a table is not normalised and, finally, normalise the table.

A sound way to test whether a proposed Primary Key actually exists as such is to see if a particular data set making up the proposed key occurs more than once in the table. Thus, those – not many, admittedly – who thought that PlotNo was the Primary Key, upon registering the fact that particular PlotNos, such as 2455, occur more than once, would have realised that their conjecture was wrong.

Although the question, as worded, did not require minimum functional dependencies, it is always good practice to provide them. Thus, if A determines B, and A plus C also determine B – adding in the C just obscures the central fact being asked for. Not all candidates were clear about this.

Providing examples of possible anomalies from an unnormalised table always causes difficulties for some candidates who are not clear about exactly what a normalisation anomaly is. For example, although it is correct to assert that, by deleting all tuples with a given plot number, we lose information about that plot's size, it is incorrect to assert that this amounts to an anomaly. It is only if we can lose information which is not directly associated with a particular plot – say, the phone number of an inspector – that we have an anomaly. One should think of an anomaly as a possible 'unintended consequence'.

Most candidates successfully normalised the table, although a few unnecessarily duplicated one or more columns in the resulting relations. (Duplicated columns are not, though, necessarily wrong by themselves. They are only wrong where they repeat information.)

The next part of the question was about alternate choices of Primary Keys for an example relation, and most candidates saw that a too narrow definition of the key would prevent the addition of wanted tuples, and that a too broad definition would allow the addition of unwanted tuples.

The last part of the question showed candidates a relation which conformed to the rules for Boyce-Codd Normal Form – 'let all determinants be candidate keys' – and yet was an unsatisfactory design. This part of the question generated a surprisingly high rate of confused answers, although the solution was simple. Students should make sure they understand 'Fourth Normal Form', which is what this part of the question was about.

Question 3

This question was a fairly routine assessment requiring essay-style answers concerning some basic database ideas: vertical fragmentation, data dictionaries, data replication and replication independence in distributed databases and query optimisation. Those who chose to answer this question – not many – had generally prepared for it and did well, although the discussions of query optimisation tended to be rather thin. Perhaps this is because the course does not focus on the physical/implementation side of database design. No candidate mentioned that some optimisers can track frequently-run queries (that is, the same query) and can store the results of the query, so that, if the underlying data have not changed from one query to the next, it is not necessary to run the query again. This can also apply to subqueries.

Question 4

The first part of Question 4 saw candidates presented with a poorly-designed relation displaying information in multiple – rather than in just two – columns. (This poor design is typical of those proposed by beginners who do not yet fully understand the relational concept.) Most candidates understood the problem and proposed a better design.

The next part of this question was perhaps the easiest section of the entire examination: definitions of basic relational ideas were required. The only weaknesses here were where some candidates used (a variant of) a term to be defined in its own definition. In other words, it is not enlightening to say that a 'determinant' 'determines' data; or that 'functional dependency' means that one value is functionally dependent on another.

The last part of this question required candidates to discuss situations which might make designers consider alternatives to the relational model. Good responses here mentioned one or more of the problems of entity types with few shared attributes (the hardware catalogue problem), or the 'big data' problem, or the 'documents' problem. Alternatives to the relational model are not explored in depth in this course; so, an extensive essay was not required – only proof that candidates were aware of the types of situations for which the relational model was possibly not a good fit.

Question 5

This question tested candidates' knowledge of basic SQL by providing simple relations and requiring candidates to frame SQL queries to answer questions about them.

Almost all candidates attempting this question performed adequately, although the usual confusions arose, as they do every year: a few candidates confused SUM and COUNT, others mixed up WHERE with HAVING and yet others equated GROUP BY with ORDER BY.

The final section asked candidates how they would test a proposed query that they had written. Good answers proposed taking—or creating – a subset of the data against which the query would finally be run and seeing if it was correct on the small set – which could be tested 'by hand'. Superior answers – probably drawing upon experience with similar problems in programming – mentioned running the query against 'extreme' or 'borderline' examples such as data sets that would return no tuples or, alternatively, all tuples. However, it was clear that many candidates had not really previously considered this important question, although the testing of coursework answers should have raised the issue.

Summary/Conclusion

Candidates reading this report and looking at past examination papers will note that there are few 'surprises' in the examinations. If you work though the course material as outlined in the Database systems subject guide, complete the coursework assignments, look over past examination papers, and take engage with the online forum, you will find that the examination is one in which you should be able to do well.