Examiners' commentaries 2016–17

CO2209 Database systems – Zone A

General remarks

The most outstanding feature of the examination this year, from the viewpoint of those taking it, was its length. Even very able students reported that they had just enough time to finish. Examiners struggle to balance, on the one hand, making questions clear, which may require extended explanatory exposition and, on the other, making them succinct. This dilemma is compounded by the fact that questions must be clear to all candidates. We will continue to work to get this balance right next year.

Despite its length, the overall performance of candidates on the examination was satisfactory, and not significantly different from that of previous years with respect to passing grades, although there were slightly fewer grades at the highest end of the distribution than is usually the case.

The usual general conclusions and admonitions are still valid: anyone taking the examination without undertaking serious revision is wasting their time. Unfortunately, there were a few candidates who seemed to have done no revision whatsoever. In this category, there were some imaginative attempts at answering questions, but the intellectual energy thus displayed would have been put to far more productive use by at least a few hours spent working through the subject guide. Examiners always try to give the benefit of the doubt to candidates whose papers are on the pass/fail borderline, but there were some very weak attempts, as noted above, bringing the overall performance down.

It should also be noted that more candidates could have taken advantage of the VLE's course discussion forum. In the weeks before the examination, students posted proposed solutions to previous years' examination questions on the forum, mainly about Entity/Relationship diagrams, Functional Dependencies and Normalization. Candidates discussed these topics in some detail, and the VLE tutor provided extended explanations. Hopefully, candidates who did not post questions viewed the discussions, but more than a few answers in the examination showed that many did not, or did not benefit from it if they did.

Comments on specific questions

Question 1

The first five parts (A. to E.) presented candidates with questions testing their knowledge of concurrent access to a database and methods allowing its recovery when processes taking place in volatile memory fail. A few candidates interpreted 'database system' to mean 'ad hoc user', and gave answers that addressed the question of data integrity, that is, how a database system can limit the number of invalid entries users may try to make. However, that was not the question being asked.

Most candidates gave satisfactory answers, and a few received full or almost-full marks on this question. Sadly, the number of full or almost-full mark answers was balanced by a number at the other end of the scale, revealing a complete lack of revision for this topic.

In addition to concurrency and recovery, (F.) this question also examined the role of the data dictionary, and (G.) the difference between 'security' and 'integrity'. Many responses to the first of these were adequate, but missed obtaining full marks because of their sparseness. When learning topics which have, say, five examples of their application, you should strive to cover all, or most, of them, as this can make the difference between gaining, say, three marks, versus gaining five.

A few candidates did not understand the difference between security and integrity. These words sound similar, because they both refer to desirable aspects of a database, but their basic meaning is different. A database can be secure, but hold wrong data. It can hold correct data, but be insecure.

Most candidates gave 'views' as an example of security-maintaining mechanisms. This is correct, but good answers went on to add other measures, such as passwords, and encryption of backup copies and of data transmitted over a network.

The subjects examined in Question 1 are 'revision-friendly' subjects because they generally enable definite answers to a limited number of questions. When preparing for the examination, you should make a list of the main points needed in an answer to give yourself a high probability of earning most of the marks. However, this requires energetic, repeated revision. A quick scan of notes on this topic will be better than nothing but won't be enough to get the 20 or more marks available if you make a detailed study of all the key points.

Question 2

This question presented a use case for which an Entity/Relationship (E/R) diagram was required (A.). It was then necessary to turn this diagram into a relational schema (B.).

A large number of candidates had a fundamental problem with this question. Their difficulty was in not seeing the 'type-vs-instance' problem embedded within it. They were presented with an airline which flew regularly scheduled flights from one city to another. There were two entity types here: the abstract flight, which had attributes such as origin, destination and departure time, on the one hand; and the set of concrete flights on the other, each of which had additional attributes, including pilot, flight length and relationships to passengers.

Few candidates grasped this distinction, which led to much confusion in the E/R diagram and the subsequent relational schema. Students preparing for next year's examination must make sure that they study this sort of distinction closely.

As always in proposing relational schema, many candidates ran into trouble when designating Primary Keys. It cannot be emphasised too often that the Primary Key is not the 'most important' attribute, but that attribute or combination of attributes that makes each tuple unique. Although this seems obvious, the reality is that many students find it a hard concept to grasp – yet it is at the heart of relational design.

Question 3

This question covered several topics. Its first section (A.) displayed an Entity/Relationship diagram purporting to describe an example of employees working for branches of a property-managing firm, who were responsible for particular apartments. But in fact, it was a classic Chasm Trap, as most candidates recognised. Students preparing for next year's exam must make sure they study and understand the differences between

Chasm, Fan and Connection traps.

The next part of this question (B.) was simple: define four basic terms. Most candidates achieved full marks here, although there were a few exceptions who appeared not to have done any reading before taking the examination.

The final part of Question 3 (C.) was more expansive, and attracted 3/5 of the marks. Candidates attempting it had to discuss how the world of databases has changed since the relational model was first proposed nearly 50 years ago (C.1), and what alternative data models have evolved to deal with these changes (C.2). A question such as this necessarily rewards those who, through personal experience and/or extensive reading, have encountered the world of concrete applications. Answers were therefore of uneven quality, with some being very basic, and a few reflecting extensive experience of NoSQL database systems. Most answers discussed the problem of document-based databases, where the structure is not known in advance, and these answers received most of the available marks.

Students aiming to achieve above-average marks on examination questions (as opposed to 'factual' questions, or questions that simply require good understanding of a relatively 'closed' topic such as 'What is deadlock and how can it be dealt with?') are advised to spend an evening or two reading around such topics. Although this is an introductory course that focuses on the basics, the subject guide includes broader topics such as alternative data models, optimisation and security, which provide the basis for extended essays. A good way to prepare to write essays that attract maximum marks is to read beyond the subject guide, and look at articles (or even YouTube videos) covering recent developments in these fields, and be prepared to refer to them.

Question 4

This was the 'normalization question'. It described a situation that required its facts to be recorded in a database, and then presented a relation which recorded these facts. The first part of the question (A.) asked for the Functional Dependencies in the table to be listed. Most candidates could do this, although a few stumbled. The most common error was to think that a Functional Dependency was a synonym for a Dependency, that is, a connection between two data items. Those who recalled the elementary definition of a mathematical function, as opposed to a relation, were probably at an advantage here, which suggests that some revision of this simple mathematical idea would be helpful when preparing for the examination.

Then came the usual question (B.) about insertion, deletion and update anomalies. Perhaps the most predictable, annually repeated errors occurred here: many candidates assumed that any sort of error occurring during insertion, deletion or update must be an insertion, deletion or update anomaly, in the sense in which this term is used in database theory. So, for example, if I update a tuple by adding false information, this is indeed an update *error*, but it is not an update *anomaly* as the term is used. If I carelessly delete a tuple, I may lose information that I wanted to keep, but only the loss of certain information within that tuple is a deletion anomaly. This is a central topic in database theory and likely to appear, in one form or another, so students preparing for the next examination should make certain that they understand what insertion, deletion and update anomalies are.

The next section (C.) required candidates to identify partial and transitive dependencies in the original relation. Most answers showed comprehension of this relatively easy question, but a worrying number demonstrated a complete absence of understanding, as if no revision at all had been undertaken.

The last part of this question (D.) required the original, unnormalized table to be recast into normalized tables. Those who understood the previous parts of this question were also able to do this part. Those who didn't, couldn't.

As a general comment on Question 4, it should be noted that more than any other question, it tended towards a bimodal distribution of marks: a significant number of them above 20, and a significant number below 10. This is consistent with the assumption that Functional Dependency and normalization make up a single subject: understand it, and you understand its various manifestations.

Question 5

The last question was the 'query question'. It presented some relatively simple tasks involving the extraction of information from a set of tables, and (A.) asked candidates to construct SQL expressions to get that information. Although there was a sprinkling of hopelessly wrong responses, most candidates did well on this part of the question.

Along with the second part (B.) about query optimisation, it was probably the best-answered question on the exam, despite the last, five-mark part of the question being so difficult that almost no one answered it correctly.

The last, difficult 'differentiating' part (C.) asked about the problems implementing a 'family tree' database: this problem is mentioned twice in the subject guide, but evidently did not receive the attention of revisers.

Where mistakes involving SQL were made, they were for the most part the usual ones: mistaking GROUP BY for ORDER BY, not understanding how to use EXCEPT as a set operator for 'not any' type questions, and not understanding the special status of NULL which makes it inappropriate as the operand of a comparison operator.

Examiners' commentaries 2016–17

CO2209 Database systems – Zone B

General remarks

Both versions of the exam this year appeared to be longer than is normally the case, with some candidates reporting that they had just enough time to finish. This was probably due to the inclusion of more questions than usual that required an extensive exposition. The problem the examiners face is that the more succinct a question is, the wider latitude there is for ambiguity and thus misinterpretation. A balance has to be struck between adequately presenting information and not overloading candidates' reading requirements. We will continue to work to get this balance right next year.

In general, results were in line with previous years: there were a few papers where the candidates were clearly just fulfilling requirements to show up, with results that could have been obtained by someone who had never studied databases at all. There were a few others where a little more revision would have taken the candidate over the pass-mark line – examiners do their best to give candidates the benefit of the doubt, but there were some very weak attempts, as noted above. The majority of candidates passed well, although there were slightly fewer grades at the highest end of the distribution than is usually the case.

This year saw extensive use of the course discussion forum for examination preparation, as several students posted their proposed solutions to past paper questions – mainly the Entity/Relationship diagram questions. Candidates had the chance to post detailed questions about the proposed solutions and a good deal of discussion ensued, which hopefully aided candidates in preparation for the examination. Students preparing for next year's examination should do the same, and the earlier, the better.

Comments on specific questions

Question 1

This was the 'query question'. For the first part (A.), candidates were required to write the SQL to extract information from the tables they were given. Most were able to do so. Where SQL errors occurred, they were familiar ones: use of '=' instead of 'IS' for the NULL entry, which is not a value and cannot be compared in the way a number or blank space can.

Some candidates revealed confusion about the proper use of set operators, writing, for example 'IN' ('2017') instead of using the IN operator against a set of values from the year 2017. Then there were the usual misconceptions which confused ORDER BY and GROUP BY, and WHERE and HAVING. A few candidates confused COUNT and SUM.

Students preparing for next year's examination should pay special attention to GROUP BY and HAVING. There are many sites on the internet where it is possible to practise, in real time, using these and other SQL statements. Learning how to use SQL, as opposed to learning about factual subjects such as optimisation, is not something that is best done by passive

revision. It is far more effective to try to solve problems and get answers in real time.

The middle part of this question (B.) asked about indexes and their use. Almost all candidates understood that an index can speed up (certain types of) access to a relation. There was some confusion about just how an index was represented – the kind of data structure/algorithm it might be (there is more than one option). Indexes and optimisation are always very likely to be included in a database exam, and you should be sure you can write sensibly about them.

The final part of the question (C.), the 'alpha' discriminator, required a discussion of how a relational database system might deal with a 'family tree' structure – in which each tuple (except the first one or two) was a 'descendant' of previous tuples. This proved a difficult topic – few candidates had good responses, even though it appears explicitly in the subject guide, and thus may be examined.

Question 2

Question 2 presented candidates with an unnormalized table (A.) and asked them to identify the Functional Dependencies in it, (B.) to use it to illustrate the classic 'anomalies' associated with unnormalized tables, (C.) to identify partial and transitive dependencies, (D.) to split the table into normalized tables, (E.) and finally to consider a table that violated none of the Boyce-Codd criteria for normalization, and yet which still required splitting in order to be properly normalized.

The majority of candidates gained at least half the marks on this rather standard question. The greatest loss of marks came when trying to answer the 'anomalies' question – year after year, many candidates assume that 'anomaly' means 'something wrong'. They propose the insertion of wrong data as an insertion anomaly, the deletion of desired data as a deletion anomaly, and incorrect updates as an update anomaly. But an 'anomaly', as this rather unusual word is used in the database world, is a specific type of error, and many of the proposed errors were other kinds of error. There are dozens of example cases on the internet of just what each kind of database 'anomaly' is, and students working for the next examination are advised to read widely until they are confident about the distinction.

A surprising number of candidates did not understand what a transitive or a partial dependency is. This is a very easy concept and everyone should have been able to gain all the marks associated with this part of the examination. These concepts usually appear in the context of taking a relation from First Normal Form to Third, so perhaps appearing out of this immediate context was a problem. In any case, these ideas are very likely to appear on any basic database examination, and students *must* be ready for them.

The penultimate part of this question (D.) required candidates to normalize the original relation. Where Functional Dependencies had been correctly identified, this was an almost mechanical process, and most had correct solutions. A few correct relations had their Primary Keys incorrectly identified, and some candidates were evidently unaware of the meaning of 'extension', or overlooked the requirement to supply one, and did not gain full marks.

The final part covered 'multi-valued dependencies'. Answering this question should be easy, as the problem and the solution are obvious. But there were quite a few bad answers here, mainly when it came to identifying the Primary Keys of the new relations. You should remember: the attribute or attributes designated as the Primary Key make a tuple unique. No two tuples can have the same Primary Key. When you tentatively assign a Primary Key to a table,

check to see if this requirement holds. Inputting the data – the 'extension' – is a good way to do this, even if it is not required.

Question 3

This question was extremely unpopular, and few candidates attempted it. Those who did, as a rule, did not do well, with very few candidates getting more than half the marks. Perhaps this was because it was not a 'theme' question, where most of the parts of the question are about different aspects of the same general topic, but covered a range of topics. Perhaps the fact that it did not specifically relate to the coursework led to most candidates deciding that it was the question to skip.

Part one of this question (A.) proposed an Entity/Relationship diagram which supposedly represented relationships among a Branch of a company, Staff Members, and Apartments they oversaw. However, the diagram as presented was a classic Fan Trap and needed to be redrawn to allow the transition from Branch to Staff Member to Apartment, and back again, to be made.

The second part of the question (B.) asked about Brewster's Conjecture (or the CAP Theorem). This is a straightforward bookwork question that should have been easy, but some candidates struggled.

The next part of the question (C.) asked about document databases – again, a relatively simple question which was generally well answered.

The next part of the question (D.) was a simple, often-asked reference to horizontal fragmentation in a distributed database, and it too was well answered.

Question 4

This was the 'Entity/Relationship (E/R) question', and most candidates gained most of the marks from it. It described a ferry company with regular sailings, involving various entity types such as ships and their pilots and passengers. (A.) Candidates had to draw up an appropriate E/R diagram, and (B.) design a normalized relational schema which captured the required information. Although few marks were lost if candidates did not grasp the difference between an abstract scheduled voyage, and its materialisation as daily sailings, several candidates didn't understand this important distinction, and tended to elide the difference between them. However, most of the other requirements were met by most candidates. The proposed relational schemas of many candidates suffered from carelessly proposed Primary Keys. One hint here: wherever a date (or other time-specifier) appears in a relation, it is quite possibly part of the Primary Key, being a unique identifier. Some candidates failed to understand this, and proposed Primary Keys that could not have distinguished one sailing from another.

Question 5

The first two parts of this question (A. and B.) focused on the physical side of database processing, asking about transactions and the ACID property, the meanings of logs, buffers, and checkpoints, and how these are used to recover from a system crash. Most candidates had evidently prepared well for this topic.

The next part of the question (C.) asked for definitions of five standard vocabulary terms from database theory. A common weakness here was to reuse a term in its own definition – for example, explaining that a 'determinant' 'determines' a data item. A simple example may help – in the

case mentioned here, it would be adequate to simply state that if for a given item 'A', there can be one and only one 'B', then 'A' is a determinant of 'B'.

The last section of Question 5 (D.) asked for the definition and advantages and disadvantages of a 'distributed database system'. Again, a very straightforward question which most candidates had evidently prepared for. Some candidates were not clear on its disadvantages – perhaps this reflects a disposition for remembering only positive things, or the well-known psychological tendency to remember items at the beginning of a list better than items at the end. However, students preparing for next year's examination must be prepared to cover the entire range of points to gain as many marks as possible.