Coursework commentaries 2016–17

CO2209 Database systems – Coursework assignment 1

General remarks

Most submitted courseworks were satisfactory, with the average mark for the whole group being above 80 per cent. Two students submitted courseworks intended for a different course: please check you have submitted the correct file. The most prominent mark-losing mishap was to fail to submit anything for particular sections of the coursework. Where a section required serious and extended intellectual effort, such as forming certain SQL queries, this was perhaps understandable, but many nonsubmitted sections involved tasks which were totally routine, such as finding a website, and could have been tackled by anyone with even basic knowledge of the subject. The reasons for their omission remain a mystery. There were a few submissions which attracted almost perfect marks for the first 75 per cent of the questions, and then nothing for the remaining few, suggesting that coursework was left until the 'last minute'. 'Just-in-time' delivery can be a valid concept for well-oiled supply chains but is a gamble when it comes to coursework.

Comments on specific questions

Part A

This part of Coursework 1 required students to download and install the MySQL database management system. In previous years, a few students have encountered problems doing this, and this year was no different. One student had to install a clone of MySQL, MariaDB. If MySQL was installed previously and then uninstalled, this seems to create the potential for problems involving port numbers. In one case, a personal firewall prevented the installation of MySQL and had to be unblocked.

In any case, downloading and installation should occur as early as possible, so that if problems are encountered, there is enough time to sort them out. Note that the CO2209 discussion forum is available to help with this. The worst thing that can happen is to leave doing the coursework until just a few days before the due date, and then to encounter MySQL installation problems which take two or three days to resolve. It is possible to roughly estimate the time all other parts of the coursework will take (an estimate of the time needed is supplied with each coursework part), but problems with downloading and installing a piece of software are not predictable.

Having installed a database management system, students were required to write a brief report on their own database experience, or if they had none, a report on any problems encountered installing MySQL, or if this was problem-free, a summary of the MySQL manual based on its table of contents. By design, this section of the coursework was supposed to get students started with some easily obtainable marks and most were successful.

Part B

Once MySQL was successfully installed, students had to create and populate the sort of 'toy database' found in textbooks, just two tables, each with just a few tuples, and demonstrate that they had done this using MySQL's feature for writing commands to a file. Almost everyone got full marks for this, with the exception of two otherwise very able students who seemed to have misunderstood the requirements and omitted to do this part of the coursework. (A note to those preparing for next year's assignment: this course enrolls students with a wide range of ability and knowledge, from absolute beginners to people with years of practical database experience. It is not feasible to design distinct coursework for each student based on their previous knowledge, so even the most experienced student must do the very simple initial exercises.)

Part C

Students then had to draw a simple entity-relationship (E/R) diagram to 'illustrate' the database they had created; the reverse of the usual (notional) process, which is supposed to start with the E/R diagram and go from there to an actual relational implementation. Marks here were mixed; most submissions got the full five marks, but more than a few fell short, giving the final average score for this section to be approximately four marks. One point worth noting for anyone required to submit an E/R diagram in future: look at the subject guide example and use it; a UML diagram is not the same thing as an E/R diagram, even though the information conveyed by each overlaps a great deal.

Part D

This assignment was to go online and locate a forum dealing with database issues, find a good question and the answers to it, and reproduce them for the coursework. Almost everyone did this and got the available marks.

Part E

This was the first section of the coursework which was not routine. It was a series of questions requiring the extraction of information from the 'toy' database via SQL statements. Some general comments: SQL is not entirely relational in its operations. In particular, queries in SQL do not return relations, they return 'tables': this means that they can have duplicate tuples which are not permitted in relations. Thus, the keyword DISTINCT should always be used after SELECT, especially where aggregate functions like COUNT and SUM are involved. It should also be noted that the latter two functions are quite different, and are frequently confused, as was the case again this year. Other mistakes were to fail to number queries, and to fail to include the results of queries. A clear example of what an answer should look like was given on the coursework itself so these errors were the result of inattention.

E2 was the most often incorrectly answered query: many students submitted SQL statements that counted the number of times the product had been ordered, rather than the number of customers who had ordered it. (In other words, if only one customer had ever ordered this product, but had placed 100 orders for it over time, a correct answer would have been '1', not '100'). A common mistake on E8 was to, in effect, use OR instead of AND as the logical operator.

The last two queries were the ones most often missed. In a small database, it is possible to check the results of a query 'by hand'. Does your SQL produce the same results that you would get if you did your query 'by hand'? Every student should check every SQL query when possible, and it is certainly possible with a 'toy' database such as that presented in Coursework 1.

Part F

This was a simple question about functional dependencies and was generally correctly answered. The idea of functional dependency is at the heart of relational database design, so the incorrect answers that appeared here, while only losing one or two marks, probably signaled further problems along the way.

Part G

This was deceptively simple. It required definitions and/or explanations for three terms which are critical for understanding relational databases. Most students got full marks, but a significant number stumbled over one or two of the three. In particular, the question, "What does 'A determines B' mean?" proved tricky for some. An adequate answer is that for any given instance of 'A', there is but one 'B', but this is perhaps too abstract, since a number of submitted answers did not make this point. A particular misconception is the idea that given a value for 'A', we can find 'B'. This is not completely wrong, but it does not distinguish that there is only one 'B'. Thus in a complete database of family relationships, given a mother's identifying value, we can find those of all her children, and vice versa. But only the 'child' ID 'determines' the mother's.

Part H

This was the most difficult of all parts of the coursework. It requested a new relational schema to accommodate a new arrangement of employee responsibilities in the original database. Where a product had been the responsibility of one employee, now it could be the responsibility of more than one. This simple change – in the relational system – required that information accommodated in the original product table be moved from there into a brand new table. This addresses the main problem that beginners find with the relational data model, and it was thus no surprise that many students, who were new to the relational model, had difficulty finding the right solution. The average mark here was 6.6 out of 10. Although the basic concepts of the relational model are not inherently complex, they are alien. Thinking relationally could be compared to having to learn to write with your other hand: it is easy to describe, but takes a lot of practice to master. The subject guide has excellent descriptions and examples of the relational model and you should take advantage of it.

Part I

This part required students to leave the 'relational ranch' and write a short essay on 'NoSQL' database systems. The temptation to submit cut-and-paste extracts from the wealth of material available on this subject was too great for some. However, those who submitted essays were rewarded for their efforts, and the knowledge they gained was useful for the examination at the end of the year (as was most of the rest of the coursework).

This is an appropriate note on which to close this report, especially for those reading it for guidance on how to succeed in this course.

Not only does the coursework mark contribute to your final mark, but much of the material you work through will also appear in some form in the examination. Thus, the effort you put into your coursework is doubly important.

Coursework commentaries 2016–17

CO2209 Database systems – Coursework assignment 2

General remarks

This coursework saw several perfect or near-perfect scores, and the average mark was just a little above that for the previous coursework (slightly over 80 per cent). The spread of marks was much narrower, with very few achieving less than 60 marks. This may have reflected the growing knowledge of those who submitted courseworks, or be an artifact of the number and types of questions to be completed; it is perhaps significant that there were fewer 'No Attempts' on questions.

Comments on specific questions

The coursework began with the requirement to download a database consisting of many tables, some of a significant size.

Part A1

For this part of the coursework students reported if they had any problems downloading the Mondial database and some did. Anyone using a Linux-based system found that the question of its case-sensitivity arose, which required renaming tables or resetting a system parameter. Other problems involved using MySQL Workbench which automatically limits the number of rows returned in a query unless told to do otherwise, and problems downloading the large Mondial dataset (solved by downloading it a part at a time). Workbench caused other problems as well. It appears that students who worked directly from the command line, accessing the downloaded 'INSERT' statements (of which there are approximately 20,000) via a simple text editor, had far fewer problems than those relying on Workbench.

Part A2

Part (a) of this question was largely an easy fill-in-the-blanks question, taking students through the commands which are useful to get information about each relation, and to show the size of the database as a whole.

Part (b) was more theoretical. The part of this question that was least likely to be answered correctly concerned issues of cardinality (how many tuples a relation has), degree (how many columns it has) and its overall size in the database (how much space it takes up in permanent memory). Most students realized – with an actual example in front of them – that the overall size of a database is dependent not just on how many rows and columns it has, but also on the actual datatype held by it. One relation could have a higher cardinality and a larger degree than a second one, but be smaller because its data was, for example, two-byte integers, whereas the other database could hold 255-byte strings.

Part A3

This consisted of six SQL queries on the Mondial database. These proved a challenge for many. Only the second query had an average mark that equaled the maximum mark: all the other queries had enough wrong answers to pull their average down to four or even to three for A3 (d)–(f). A couple of submitted queries were obvious fakes; the attribute names were spelled incorrectly, which should have caused an error, and yet results were presented. MySQL does not yet have a spelling correction function! For A3 (e) and (f), an error that occurred several times was to use the wrong relation in the query, for example, 'Located' instead of 'geo_river'. The lesson here is obvious: become familiar with the 'semantics' (meaning) of the database you must query. For queries A3 (c) and (d) a common mistake – **the** most common in fact – was overlooking TYPE='member' for the World Trade Organization, thus counting 'observers' and 'applicants' too. Again, the moral of the story is to study the meaning of each relation before attempting to query it.

Part A4

This had two parts, each designed to familiarize students with sources of online help. First, the students had to go online to find a description of how to use MySQL's GROUP CONCAT feature (a nonstandard 'denormalizing' feature useful for creating user-friendly VIEWs); second, they had to find videos on 'Big Data'. The latter should have been helpful for the examination. Almost everyone completed this exercise successfully, although there were a few puzzling omissions. Watching a video was required for this assignment, and yet some students who answered the harder questions without difficulty did not complete this part of the assignment. Perhaps the students who lost marks here put off this part of the assignment until the last minute, and then found that they did not have the time to do it.

Part B1

Here students had to draw an entity-relationship (E/R) diagram which captured the key information presented in a description of a school of beauty therapy, with lecturers, courses, modules and students. Errors here included eliding the difference between 'Can Teach' and 'Does Teach', adding a redundant relationship between lecturer and student (they are indirectly related via module and have no direct relationship), and missing out relationships altogether (the most common mistake). Although the standard approach says to go from verbal description to E/R diagram to relational schema, many students will find it easier to switch stages two and three, because designing an E/R diagram involves engaging with abstractions with which students may not be familiar. This was proven in practice by some students who submitted a relational schema which recorded information that was missing from their E/R diagram.

Part B2

In this part of the coursework students had to present a relational schema which allowed the recording of information presented at the beginning of Part B. Each 'Entity Type' needed its own 'Master' relation, for recording information exclusively about itself (the 'attributes' of the E/R diagram), and then the relationships among entity types required seven more relations. This was a challenging assignment and, even with generous marking, the average mark was 10 out of 15 possible marks.

The most common errors here were to design relations which were not in at least third normal form, and to designate incorrect primary keys.