Coursework commentary 2017–2018

CO2209 Database Systems

Coursework assignment 1

General remarks

Coursework 1 was completed by almost all students. A few submissions showed evidence of a very late start, namely, non-completion of fairly easy parts of the coursework assignment resulting in unnecessary loss of marks. However, the overall the standard was good. A few students got 100%, and the majority were comfortably above 70%. There were very few outright failing marks, and these were mainly because of 'no attempt' submissions, especially the SQL queries.

Candidates should try to make an early start on coursework assignments. In particular, it is well worth familiarising yourself with the whole coursework assignment as early as possible, before beginning to work on it. This will allow time to take advantage of the course discussion board to ask for clarification of anything that is not clear.

A few candidates did not follow instructions, submitting several separate files instead of one PDF file. Doing this relies on the examiners taking the trouble to move back and forth between two or even three different files linking questions with answers, which is cumbersome. Other candidates did not number their queries or did not, as asked, include the natural language question to which the query was an answer. These omissions will have made their coursework assignment less useful for revision.

Part A

This part of Coursework 1 required candidates to download and install the Open Source database package MySQL and to report on their experience of doing so. A few candidates have problems doing this every year, but this year the number of candidates experiencing difficulty increased significantly. Some typical comments were:

"Installing MySql is a very tedious process which it should not be since it is so widely used and it is an open-sourced with so many people contributing to it."

"The MySQL manual while informative, can be confusing to new users like me when trying to download the correct packages. The instructions given were simply to download the package individually and I found that installing MySQL via the MySQL installer for windows was easier and faster."

We will take account of these issues in future coursework assignment instructions.

Part B

Candidates were required to use the MySQL manual to find the answers to some simple how-to questions. Almost every candidate got these right.

Part C

This section described a business case for which an Entity/Relationship diagram had to be drawn up. This is always a subject that many beginners find puzzling, so it was not surprising to see MODEL as an attribute instead of an Entity Type. Not seeing that 'authorized' was a relationship on its own was an error that appeared several times. There were more than a few superfluous relationships in some E/R diagrams, but, interestingly, these usually *didn't* show up in the actual implementations. This suggests that first attempting an abstract Entity/Relationship analysis, and deriving relations from it, does not always suit everyone: going from the abstract to the concrete is the reverse of normal thinking.

Part D

Here candidates were given of a set of exercises designed to teach the art of detecting functional dependencies in related data.

A typical error was 'Partcode + Name => Weight ' instead of Partcode alone being the determinant. This was not an uncommon fallacy, probably resulting from thinking along the lines of: 'a Part' has a code and a name that identify it; and it has a property, 'weight'. Some candidates evidently did not think that the determinant in a Functional Dependency could be compound. Some did not understand the idea of 'Left-irreducibility', which just means that the determinant in a functional dependency should be as simple as possible, with no 'extra' attributes.

Part E

Part E required a relational schema to be developed from the situation described in part C, using their E/R diagram, or to be derived from the description directly. Errors here included not normalizing the relations, or not showing the Primary Key. Some candidates had an incorrect E/R diagram – missing out relationships or adding spurious ones – but a correct relational schema.

Part F

Building on from the previous section, the relational schema developed in Part E was actually implemented with sample data. Some candidates had an incorrect relational schema, but a correct implementation.

Part G

Candidates were asked to modify the database in Part F while violating built-in constraints. This was generally done well, since if the database was set up correctly, the proposed modifications generated an error message which then simply had to be copied down.

Part H

Candidates were presented with twenty natural language queries on the database, to be answered by constructing SQL statements. Some of the most common errors here were:

- not submitting the output of the query, as required
- 'hard-wiring' part of the answer into the query itself
- breaking what should be a single SQL expression up into several queries; confusing 'name' with 'number'

- using GROUP BY where ORDER BY was wanted
- using COUNT where SUM was required

Query H6 was interpreted by several candidates to mean 'at least one' rather than 'all'; perhaps the thinking here was 'at least one of the models has been distributed by at least one of this group, therefore the group has distributed all models'.

One way to get the wrong data in response to a query is to enter the wrong data into the relations in the first place, and several candidates did this. Double checking your data entry is never a waste of time.

In summary

To get maximum marks on the coursework assignments, you should start early, use the course discussion board to resolve any apparent ambiguities in the questions, follow instructions closely, watch out for common misconceptions in using SQL and double-check your data entry. Even if some of the concepts involved in the assignment are difficult, there is no good reason to lose marks through carelessness.

Coursework commentary 2017–2018

CO2209 Database Systems

Coursework assignment 2

General remarks

This coursework assignment was not quite as long as the first one, but seemed to prove a bit more challenging to many. Results were acceptable, though there were fewer top marks and a greater spread of marks overall.

Part A

The first section required candidates to download the 'Mondial Database', a compendium of 'world-facts' as a set of relational tables that allows users to get experience with a database larger than the typical 'toy databases' of the textbooks. Candidates then had to determine the total size of the database, and answer questions on the relationship between total size, and cardinality and degree. Candidates who paid close attention to the suggested examples found that it was not always the case that, of two tables, the one with the largest degree and largest cardinality is also the largest in terms of data. But some candidates did not see this. In some cases, the problem was failing to understand the difference between 'could' and 'must'. Close attention to the actual figures revealed by following instructions would have shown the, perhaps, counter-intuitive reality.

The second section of this part consisted of ten questions about the information stored in the Mondial database. Because this is a large database, candidates were requested to add LIMIT 10 to their queries, which would have restricted the returned data to 10 instances – but some did not. Another common mistake was when a candidate submitted SQL statements but not the data those statements which would have generated. The request for the data is for your benefit as well as that of the marker, because examining the data helps to locate where a query may have gone wrong. For example, if the COUNT function was used on queries and the SQL was wrong, this function always returned a count of zero. This should have been a clue that something was wrong.

The usual SQL puzzlers appeared and tripped up some candidates. Some misunderstood how GROUP BY works to frame a 'negative' query where we want retrieve, say, all countries which are NOT members of the World Health Organization. If a country can be a member of several organizations at once, then the 'not-equals' sign simply gets you countries that are members of one or more organizations which are not the World Health Organization, which does not exclude membership in that organization also.

Some minor errors were:

- returning a country *name* when a country *code* was asked for
- returning more columns than were required
- not numbering the queries

None of these errors were penalized, but they are evidence of a careless approach to work which we do not want to encourage among professionals.

Part B

In the first section of part B, candidates had to choose the proper primary key of a proposed relation.

In the second section, candidates had to re-design a relation which, while technically in Boyce-Codd Normal Form, was still a poor design.

In the third section, candidates had to identify the faults in two different design solutions to a problem, and then propose a third solution without the faults of the first two. Many misunderstandings were revealed here, and future coursework assignments will be carefully designed to make these explicit.

Finally, candidates were given an example of a proposed E/R diagram that contained a 'trap' which would lead to faulty relational design if used as a basis for creating a relation. Almost all candidates recognized the problem and proposed valid solutions for it.

Part C

The final section required candidates to research and write a short essay about 'the null controversy', and then to find and view some online videos which dealt with the topic of 'NoSQL' databases. With respect to the 'null controversy' (whether or not the 'NULL' non-value should be allowed, as it almost universally is in current practice), it was disappointing to see that a very large number of candidates simply embraced the no-nulls-allowed view of the contrarian website whose URL they were given. They therefore missed a chance to weigh up the arguments on both sides and exercise critical judgement. Just because experienced practitioners can argue convincingly for a certain case, doesn't make it automatically true. In particular, a few answers pointed to the obvious objection to dis-allowing NULLs: what do we do instead? We will take this on board, and will provide explicit links to all sides of controversial questions in future coursework assignments.