Comments Q1

This was not a difficult question, with mean = 62.6%, but had a rather high standard deviation = 21.5.

A.i This was simple recall, but answers were often off target in a multitude of ways. Perhaps the most common misunderstanding was that many students associated the notion of superkey to superclasses, which is plain wrong. A superkey is a set of attributes whose values uniquely identify each tuple in a relation. A superkey is reducible in the sense that it can lose one or more elements in its attribute set without losing its ability to uniquely identify tuples. A key is irreducible, i.e., if it loses any element of its attribute set, it ceases to be able to uniquely identify each tuple.

A.ii Many students failed to link the *logical* notion of foreign key with a corresponding *conceptual* notion, i.e., that foreign keys implement (or capture) the conceptual notion of a relationship between entities.

B.i The most common cause of loss of marks here was the failure to identify this as a GROUP BY aggregation query suing COUNT. This is straightforward SQL but some students got lost with procedural extensions, which are not needed.

B.ii Other than routine confusions about relational-algebraic syntax, this part was not plagued by systematic misunderstandings (other than, as with B.i, a tendency to stipulate unnecessary joins, which was not normally punished in the marking).

B.iii In this part, many students forgot the UPDATE … SET … construct in SQL, which is, as all of SQL, set-based. As a result, there was a tendency in these cases to resort to procedural extensions, again completely unnecessarily.

B.iv The greatest misunderstanding here was to overlook the fact that the DDL part of SQL already captures the four options required here (viz., reject, cascade delete, cascade set to null, cascade set to default).

B.v The most common problem was a suggestion to make the attribute a foreign key, when all that is needed to to declare it NOT NULL.

Q2

This turned out to be a difficult question, with mean = 45.1% and (quite high) standard deviation = 26.1.

A.i This turned to be a very hard subpart as there was an almost universal failure to recall that a claim that a functional dependency (FD) holds is a claim that the FD holds over all valid states of the database. Thus, looking at one (as given), or even many (however many), states does not substantiate (other than probabilistically) the claim than an FD holds. What one needs here is an assertion about the domain semantics: exhibiting states (one or more) can never be sufficient. Another common misunderstanding was to try and answer the question by applying Armstrong's axioms, when the question was about the given state of the database providing enough information to confirm the claim that the given FDs hold or not.

A.ii This subpart didn't lead to significant, systematic misunderstandings. Unlike (A.i) (above), in this case, a single state in which a postulated FD is violated is sufficient for us to infer that it does not hold.

B.i Many students got confused here. The relevant FDs are AB->C, AC->B and BC->A, therefore AB, AC, and BC are all the candidate keys. It follows that any choice of left-hand side (LHS) as primary key will leave out a single attribute (on the right-hand side (RHS) of the corresponding FD) that is dependent on the key, the whole key and nothing but the key, i.e, the relation will not violate BCNF for each and every possible choice among the possible candidate keys. Many students erroneously saw transitive dependencies, but, in the above FDs, since no RHS appears in any LHS, it follows that there are no transitive FDs.

B.ii Most students were lax in answering 'B and D' without being clear that they meant the set {B,D}, i.e., a single composite key as opposed to alternative keys {B} and {D}. Only the former is correct, but harsh penalties were avoided.

B.iii This subpart didn't lead to significant, systematic misunderstandings. Some of the answers were a touch too terse but, again, penalties were avoided as much as possible.

Q3

This was an easy question, it was taken by 162/188 students. The statistics were: mean = 75.4% and standard deviation = 16.7.

A. This was a generally well-answered part. The only systematic error worth noting was the failure to draw the participation constraints on relationships.

B. This was also a generally-well answered part. The most common cause of loss of marks was a failure to show, as the question asked, the step by step derivation of the relational schema. A little less often, students also preferred (for no good reason in this case) the relationship relation (rather than the foreign key) approach when mapping 1:1 and 1:N relationship types.

C. This was in general a poorly answered part. No systematic errors were spotted other than that students failed to reveal an understanding of the difference between category types and the other kinds of entity types in the extended ER model.

Q4

This was a difficult question, taken on by 25/188 students. The statistics were: mean = 38.8% and standard deviation = 19.7.

A.i Most students had no problem with this part.

A.ii Most students found this hard and came up with too many false problems. Also, those got it right rarely found more than two true mistakes.

B.i Few students recalled the state transition diagram introduced in the course unit.

B.ii Here, by far the most systematic mistake was not to *draw* the precedence graph, thereby losing marks. Listing the timelines of the two schedules was not what the question required. Several students stated conclusions about serializability but *not* backed it up with the precedence graph.

B.iii Most students got some marks here with no discernible systematic misunderstanding.

Overall, the exam has a mean of 62.6% with 17.8 standard deviation.