PGT Exam Performance Feedback
Semester 1

Comments:

Please also see PDF for this unit.

Q1.1 all three are bookwork questions, but we look for proper explanations using correct terminology; in particular, for 1.c, we indeed want to see in how far the infoset and the PSVI differ.

Q1.2 partly well answered, but far too many confused
---- different kind of statements that we can make in a schema -- e.g., what is allowed/expected -- with
---- different properties of an XML document that we can describe in a schema -- e.g., the document structure (element names and their nesting -- and its content -- e.g., the datatype of attribute values.

Q1.3 (a) was mostly answered correctly, but answers to (b) often failed to mention that, for a snippet like the one given, the use of namespaces seems over the top and some aren't used -- i.e., there is no benefit in declaring them at all.

Q1.4. was mostly answered correctly, but some failed to spot the default attribute value set in the DTD, thus returning in too few elements returned; some failed to give any explanation; some failed to return the right structure of elements.

Q1.5 a+b were mostly answered correctly, © not quite as well.

Q1.6 was the most challenging of (1) questions, and we expected students to answer both "why" and "when" (only 8 students got full marks).

2.5
Points (up to 4) were given for correct advantages and disadvantages listed. Minor errors in definition or pro/con were tolerated, but major ones could cost points.

"Internal" validation does not refer to having an internal DTD "subset" i.e. to the physical inclusion in the file. Answers centered on this generally were completely wrong (i.e., 0) though I did give a point for some arguments about physical separation (which is true for external validation and subset).

External validation more easily accommodates using multiple schemas.

Time is not a significant advantage or disadvantage of internal vs. external, esp. due to network concerns (since those apply to external subsets as well). This should be familiar as you all did plenty of external validation in the course work: the files were local, you could configure a one click move, it clearly had no major impact.

Since the distinction naturally applies to DTD contexts, namespaces cannot be an issue.

Validation modality does not affect the possible tightness or looseness of the schema. Arbitrary DTDs can be used in either mode. Both modes produce PSVIs.

Some people provided definitions (even somewhat correct ones) without listing advantages and disadvantages. This didn’t answer the question and got 0.

2.8
Some people did not understand that the question was asking to consider Xpath (a "query"
language, typically used to "retrieve" parts of a document) could be used as a "schema" language (i.e., to "check" the structure). The question did not ask for a comparison of Xpath and (e.g.,) XSLT or Xquery. Nor did it ask for a detailed description of Xpath.

Many people just gave advantages or disadvantages of Xpath (e.g., over Xquery or SAX). Where these made sense for Xpath as a query language, I tried to give points for that.

Some people claimed that Xpath couldn’t handle certain things, such as “namespaces” or “datatypes”. This is incorrect, at least in general. Xpath functions can be used for type "checking" of atomics and Xpath is certainly namespace sensitive. Xpath does not generate a PSVI, of course, and there are other differences. All other things being equal, this cost a point.

It is not true that “most” schema languages use Xpath…only Schematron does.

By and large Xpath can be "less" sensitive to changes in the document. That is, it can be used to validate particular fragments of a document even when the rest changes wildly.

Every Xpath either succeeds or returns no results. If it succeeds then it has "checked" that constraint. That was the point of using conditional expressions in class to simulate returning "IS VALID".

You don’t have to use schematron to use Xpath to validate a document. The success or failure of the query is similar to the success or failure of validating against a DTD.

2.10
Rather a lot of people had syntactically correct DTDs that modelled the document accurately. Good job. A high number of the remaining had minor, easily repaired errors and received full marks.

To test your answer, take the sample document, add the DTD internally, and see if it is valid, then to make sure it made sense (i.e., didn’t just make everything ANY). I corrected simple syntactic errors (e.g., <!ATTRIBUTE --> <!ATTLIST) for no points off. Missing information in the modeling (e.g., <!ATTLIST topic id CDATA title CDATA > (it needs a default value, i.e., #REQUIRED) would cost a point for that declaration. If the same error was repeated and was relatively minor, I only took off for the first one, e.g., in

<!ATTLIST topic id CDATA title CDATA>
<!ATTLIST person nick CDATA>

technically, both lines are wrong (missing default value), but since they are otherwise complete and the error is the same, I only marked off for the first. Compare with something like:

<!ATTLIST id value="#REQUIRED">
<!ATTLIST title value="#NMTOKENS">
<!ATTLIST person value="#NMTOKENS">

I ignore the syntax problem (value="") but there’s so much wrong (missing element, type in the first, missing element, default value in the second, missing attribute name default value in the third; the given types are implausible given the content; title’s type is wrong (NMTOKENS don’t have ‘’s.) that this lost three points.

Here’s a clear mismodelling:

<ATTLIST topic title cdata #IMPLIED "hume’s theoru of knowledge">
Even if we reconciled the #IMPLIED with the default, it’s not a plausible default (and doesn’t even match the given text).

Several people got confused about the nesting of topic and discussants, e.g.,

<!ELEMENT discussion (topic,discussants)>
Discussants is not a child of discussion, but of topic.

One possible "syntax" error was:

<!ELEMENT person>
for <!ELEMENT person EMPTY)>

I interpreted that as incorrect (-1) under the “the info is missing” rather than garbled.
I accepted this:

```xml
<!ELEMENT person (#PCDATA)>
```

though it’s really not plausible or good modeling. It *will* validate.

I gave one free typo in the element/attribute names, but took -1 for more. XML is draconian.

Several people missed the person declaration. Every element (and attribute) must have one.

2.14

The key point that most people hit is that DTDs are simpler (easier to read/write, etc.). The desire to inline the schema was accepted, but doesn’t seem that compelling. Rather fewer people pointed out that DTDs can declare entities (a very good reason).

DTDs (except for entities) can be expressed as WXS but not the reverse. There’s no reason to think that DTDs are hugely more efficient than the correspondingly expressive WXSs.

DTDs are *less* constraining that WXS. For example, a DTD can not constrain element content to be only integers, or only 3 characters long.

Some people gave conditions which *permit* the use of DTDs (e.g., no namespaces) but didn’t give a positive reason for using them.
Q 1.1 Answered well. Most students named appropriate techniques and provided descriptions. There was some occasional confusion over the details. For example, in 20 questions, the Domain Expert asks the questions -- it is the order of questions that is important. Full marks were obtained for answers providing descriptions of the procedures plus some rationale or explanation of why the procedure is as described.

Q 1.2. Answered well.

Q 1.3 Bookwork.

Q 1.4. Answers to this question were varied.

A number of answers made reference to soundness of the reasoner, which is not relevant here.

Nor is this specifically a question about OntoClean -- which is primarily a mechanism for determining whether subsumption hierarchies are appropriate.

Some answers discussed whether the use of a non-transitive direct-part-of property and a transitive part-of superproperty would solve this. This pattern is used to address issues with transitive relations -- for example being able to distinguish between asserted and inferred relationships, but would still not necessarily provide the solution to this particular problem.

The issue is that the partonomic relations represented here are different -- the relationship between my hand and me, and me and the IMG are not the same (recall the classifications from Odell and Winston), thus it's not appropriate to consider transitivity across these relations. We can get round this by introducing hierarchies of partonomic relations, e.g. component-integral, member-bunch etc., and only asserting transitivity at the appropriate level.

Q 2.1. Answered well overall. Some answers referred to transitivity or lack of transitivity. Although this is a distinction, the key distinction is that the broader/narrower hierarchies in SKOS are "looser" than those in OWL -- OWL's subsumption relationship has a formal interpretation in terms of instances, while SKOS hierarchies simply represent some notion of narrower/broader.

Q 2.2. Most answers discussed applications supporting some kind of navigation, which don't, in general require the same levels of ontological commitment that would be provided by an OWL ontology. In general, though the applications weren't describe particularly well.

Q 2.3a Few answers correctly identified the three principles here. Most said something about the use of URIs, but the principles of providing dereferenceable HTTP URIs and the provision of useful information using standards were missed by most.

Q 2.3b Few answers provided much explanation or justification for their answers. We can argue it's not Linked Data in that there are no links "out" of the data source. However, it can serve as a target or sink for links, and thus can still participate in the "Linked Data Cloud" -- but it's a dead end. Using the first three principles still allows the use of standardised infrastructure to access the information. However, not using external URIs to describe the vocabularies may make interpreting that data difficult.

Q 3.1 Bookwork question -- most answers correct.

Q 3.2 Bookwork question -- most answers correct. Some answers failed to state that we require a model such that the interpretation of the class is "non-empty".

Q 3.3. Well answered overall. Most students provided an example of an consistent ontology with an unsatisfiable class.

Q 3.4 Many completely wrong answers here. A number of answers failed to provide what was asked for in the question, which was a model/interpretation of the ontology provided.
This should consist of a domain (set) along with mappings from primitives (classes/properties/individuals) to elements and subsets of the domain. Some answers consisted of ontologies which were completely unrelated to the question, others simply translated the given ontology into a different syntax.

Feedback on individual answers refers to axioms as follows:

1. A subClassOf B
2. C equivalentTo (B and (R some D))
3. j: (A and (R some C))
4. i: (B and (R some D))
5: k: (S some A) and (all S (not B))

Where interpretations were actually given, in most cases the answers were good. Most of the interpretations were models of the axioms, although axioms 2 and 4 caused some problems.

Q 3.5 Overall, answers better than Q3.4. Most actually provided a concept definition (although there were many syntactic irregularities). A number of answers suggested D = (R some B) and (R some C), which rules out III, but means that i:D in II. We need to use a universal in the definition to preclude the R relationship to a C.

Some answers provided a subclass axiom for D. Note that this doesn't provide the desired effect as we won't be able to *recognise* i as a D. An equivalence axiom needs to be given.

Q4.1 Many good answers. Some people didn’t catch how increased expressivity might have a neutral (e.g., with a lot of syntactic sugar) or positive effect on computation time (because, e.g., allowing for a terser representation). If the answer was otherwise good, this cost a point.

Note that *lack* of expressivity can cause people to write more complex ontologies as they try to work around the lack.

I was a bit generous about a lack of examples, generally taking one point if they were entirely missing

Q4.2 To get full marks, you need to get both the connection via expressivity (positive and negative) and via sheer performance issues (i.e., if responsiveness is low, people will find the language hard to use/learn/etc.) People typically missed one of these, but got others.

Several people pointed out that lower computation complexity (because of low expressivity) could *increase* cog complexity.

Q4.3. “Practical” here refers to a complexity theory term of art; however, I did give some credit for talking about practically more generally. People did suggest that practical suggests e.g., unsound, which is not true.

Practical OWL 2 DL reasoners do *not* (necessarily) perform well on *arbitrary* input. Arbitrary input includes worst cases. They perform well on “realistic” input.

Q5.1 *Wrong* answers might never be *preferred*, but they might be tolerated in return for other benefits (e.g., performance) if they aren’t too bad. To get full marks, the answer had to point to a trade off.

Many people didn’t point out that the reasoner must be complete enough (or just complete). If we are unsound and radically incomplete we’ll get *only* wrong answers.

Some people gave the case that if we are modelling a fictional world it would be ok to use an unsound reasoner because “true” answers don’t matter. But this is incorrect. “Sound” answers are sound *relative* to the model.

Q5.2 Again, a key aspect is *getting* something for the incompleteness, e.g., performance.

Note that being incomplete does not force you to be *either* sound or unsound (though the former is more standard).
Q5.3 Most people got this right.

Q6.1 Most people got this right. A key point was to note that runtime imposes higher performance demands than development time. Another key point was to talk about how inference can lower development costs.

Q6.2-3 The key to this and the subsequent question is dealing with the “defaultness”. I.e., if it’s “built in”. Typically with these global features, they make something easy (e.g., that I have at least 4 friends from listing four of them) but makes other things hard (e.g., that I have one friend with two names).

UNA only applies to names of individuals, not to class or property names.

Since the UNA and OWA have advantages and disadvantages, it was not sufficient to make generic claims such as “It makes things easier.” You had to give some specific things it does in fact make easier.

The UNA and OWA are compatible. Many description logics have both. The OWA does not force the UNA (OWL has the first but not the second).

7.1 Most people gave good if rote answers. A few people got confused by the use of the term “ASK language”, e.g., thinking it designated boolean queries, specifically. It refers to arbitrary query languages. Both class expressions and conjunctive query can be executed in mere boolean mode.

7.2 Again, most people gave good, if rote, answers. However, a really nice answer suggested that reasoning times might be worth returning. Most people didn’t get into the rationale or the circumstances.

Several people echoed the question, e.g., “The result differs from query to query”. Obviously, what is written in the question does not enhance your answer.

It’s not a particularly useful DESCRIBE to return exactly what could have been retrieved with a SELECT. So for ?x rdfs:subClassOf ?y, it wasn’t responsive to say that it should return ?y and all its subclasses. That’s exactly what a DESCRIBE “shouldn’t” return.

---

**COMP 60611**  
**Fundamentals of Parallel and Distributed Systems**  
**Len Freeman**  
**Graham Riley**

**Comments:**

1. Parts a) and b) were completed reasonably well. Difficulties with parts c) and d) were associated with a less than complete understanding of the concurrent execution model of FSP (in terms of labelled transition systems).

2. The performance modelling question.
   a) (ii) Twice as many messages when \( P=N \), compared with the other cases.
   b) (iv) 2-D partition has twice the number of messages, but their length increases like \( O(N \sqrt{P}) \), whereas, in the 1-D partition, the length of the messages increases like \( O(NP) \).

3. Least convincing attempted solutions. Parts a) and b) suffered the difficulties described in 1. above. Part c) lacked critical analysis of the relevant issues, with little, or no, reference to the literature.

4. The queueing theory question.
   In general the correct methodology was used but a number of the marks lost were due to algebraic errors that should have been obvious.
1. The most popular question; answered by almost all the candidates. Generally the answers were good. The key points to note were as follows:

   In part ii) the timing is of the vector axpy loop ONLY. The sequential initialisation in Implementation A leads to many remote memory accesses that are not present in Implementation B.

   In part iii) all three loops are parallelisable, but the inner loop contains a reduction variable that must be addressed. Parallelisation of either the middle or the outer loop introduces load imbalance issues.

2. The only difficulties with part (a) were the clear identification of the savings resulting from the different strategies (and the associated programmer effort - little for 1) and 2) and substantial for 3)). For b) the call to ISSQUARE affects the workload and influences the most suitable scheduling technique to achieve load balance.

3. Least popular question.
   a) Variation on bookwork.
   b) Code fragment 1 readily parallelises - no dependencies. Code fragment 2 requires to loop distribution to enable parallelisation.
   c) Need to remove loop-carried dependence due to statement k=k+1.'if' statement implies triangular iteration space and load balance issues to address.

4. Generally a well done question - major difficulty was in providing a clear and succinct description for part d).

Comments:

On the whole, most students scored excellent grades. The most common source of lost marks was in application of Bayes Rule (Q2) and the understanding of ROC analysis (Q1a/Q3a).

Q1. Answered excellently by most. Common mistakes were:
   (a) Some stated "testing error". This is the same thing as generalization error, given in the question, hence not part of the answer. Additional factors are those defined by ROC analysis.
   (b) Some simply defined over-fitting rather than actually answering the question of how to control it.

Q2. (a/b) Seemed to polarize the group. Students either understood Bayes rule or did not. The answer is a simple application of the rule discussed in class.
   (c) Very few got this correct - confusing it with assumptions of having sufficient training data to learn - the correct answer is that it assumes the features are independent given the class label.

Q3. (a) Very few got this correct. Almost all mixed up the concepts of sensitivity/specificity with bias/variance. The question is about ROC analysis, hence nothing to do with bias/variance.
All students attempted questions in three sections. Their marks are roughly distributed in a normal distribution; i.e., one above 70%, nine in the interval between 50% and 70% and two below 50%. The highest mark is 74% while the lowest one is 38%.

For short questions in Sect. A, nearly all students have a good understanding of basic concepts and book knowledge in general. As a result, all students received at least eight out of 10 marks from this section. A relatively common problem appears in A4; i.e., a couple of students described the SOM algorithm details other than the behaviours or processes of SOM learning.

For essential questions in Sect. B, most of students performed well. However, some students did not describe their answers properly; they presented algorithms for a general case while questions ask them to apply appropriate algorithms to specific real-world problems. In addition, most of students failed to give a complete answer to the critical analysis question in B(c).

For comprehensive and formal analysis questions in Sect. C, most of students failed to answer C1(a) or C2(a), formal analysis questions while most of students managed to answer questions well where the book knowledge can be applied relatively straightforward. In general, students seemed to perform poorly for those questions that require formal analysis or mathematical skills and deep understanding of comprehensive knowledge for problem solving.

In summary, the overall result looks quite reasonable and accurately reflects what students actually achieved from this course unit. In particular, the distribution of examination marks is consistent with that of coursework assessment.
Question 1
In the structural CNF transformation:
the name for the top formula was not included into CNF.
Not many attempted to answer the question about Inst-Gen.

Question 2
Considerable errors in the DPLL calculations.
The formalisation problem in propositional logic (2.e) was problematic for many.

Question 3
Generally the level of explanation and justification was good.

a) generally answered very well (and much better than in previous years). Common reasons for using marks: imprecise formulations and explanation and confusing sound rule with sound calculus in iii.

b) Generally, i posed no problem. Else answers were mixed. For iii it helps to represent every man owns a car and then negate it.

c) If the first occurrence of `all x' is moved outward first and then the second occurrence of `all x' is moved outward, followed by moving negation inwards before replacing implication the number of steps is fewer. But there are many ways to do the transformation all were acceptable. Structural transformation could be used and is not wrong but is not necessary.

d) Everyone who attempted the question managed to give an interpretation with one element in which the formula is true. It then still needs to be shown that the formula is not true in any 2 element interpretation.

Question 4
Generally the level of explanation and justification was good.

a) generally answered well.

b) i) It is important to realise that if a clause contains a non-ground literal then it is wrong to just count symbols. For determining maximality of non-ground literals it needs to be considered if there is ground substitution which makes the literal maximal. See the sheet.

   ii) Mixed answers. Common mistakes:
   - forgetting that factorising is a possible inference rule to apply
   - not performing inferences on maximal literals
   - identifying the wrong literals as maximal

   iii) When answered, answers were correct.

C) Least popular question.
15 students sat the exam. All but two obtained over 50%, and one of the two was only just below. The student with the lowest overall mark performed badly on all questions and had clearly not prepared satisfactorily for the exam.

Q1: All but two students attempted this question, and generally performed well. Most lost a mark or two for not explaining the code features fully, e.g. not observing that ADR is a pseudo-instruction, not observing that the LSL #2 is scaling to word, etc.

Q2: All but four students attempted this question, again performing well, though some lost marks in the 2nd half for failing to identify the key issues in ARM-Thumb interworking.

Q3: About half the students attempted this question, with less good performance than Q1 and Q2. Few had really got to grips with cache power optimisation as described in the on-line course material (and the course book).

Q4: Again, about half the students attempted this question, with one getting full marks but the rest much poorer. Understanding of the material was muddled, and back-of-the-envelope estimation for the last part of the question was weak.

Q5: The majority of students attempted this question, with good results. It is hard to get full marks here, and few understood the subtleties of exploiting the on-chip bus for production test purposes sufficiently to avoid losing marks here.

Overall the performance this year was in line with previous years, though two students answered more than the required number of questions (but gained no more marks for so-doing), and several of the answers were waffly despite the request for concise answers in the rubric. There was still a tendency to write everything they knew about a subject rather than answer the question!

Comments: All questions had an average close to 50% of the available marks. It was clear that students had, in the main, revised rather selectively.
Feedback awaited.
Q1: Least popular question. 'Moving Holds' were not well understood (not allowing any part of character to become completely still, moving slightly in direction of motion). Confused with creating a thinking character (which is more about eye movement). The advantage of 3D over 2.5D animation is that 3D preserves shape by interpolating object transforms, not shape/vertices.

The RenderMan questions were lacking in details. MicroPolygons are formed when dicing primitives in to sub-pixel sized polygons. Use of fragment, vertex and geometry shaders on the GPU may allow implementation of RenderMan in hardware. GPU creates fragments, similar to micropolygons on which shaders can operate.

Q2: Most popular question and generally well answered. This subject has been well revised. Description of Euler angles was a little vague. Factor rotation in to independent rotations about principle axes, order should be consistent throughout operations, can combine matrices. The proof was usually well done but marks lost by not multiplying the matrix with a general \([x,y,z,1]\) point to show effect on the z component (it becomes \(-x\), hence gimbal lock). Quaternion questions were answered well. Marks were usually lost in the procedure for deciding between \(q_2\) and \(-q_2\) during interpolation. Take the dot product of \(q_1,q_2\) and if cosine of angle > 0 choose \(q_2\) otherwise choose \(-q_2\). Also, the problem with \(\text{Slerp}()\) is that it only produces constant speed rotation (looks unnatural). Alternatively it has discontinuities in the interpolation hence abrupt changes in orientation.

Q3: Second most popular question with a range of marks. A hierarchical animation system means animator doesn't have to manually apply transforms to all child object in a hierarchy. The tree diagram was mostly correct although labelling of arcs with both translation and rotation transforms was often missing. Walking down one branch gave the sequence of transformations. The description of the IK algorithm was OK but missing a few details. Marks were lost by not stating that the Jacobian relates joint angle velocities to end-effector velocity (velocity, not position/angle being the key point). \(J\) has to be recomputed each time step and may not be square (so use pseudo \(J\) in order to invert it). The description of the walk cycle tool was mixed. Most people described something similar to the tools demonstrated in the lecture (which was correct) but were a little vague. The IK technique should be applied to creature limbs and then standard animation methods or procedural methods, based on parameters such as stride and leg length, could be used to move the end-effectors.

Q4: Just under half did Q4, the other half doing Q5. Most understood the basics fairly well although about half missed specifying the priority of the flocking rules; and then how new extra rules would include themselves within this priority level scheme. Main loss of marks was in not describing the nomenclature for their diagrams and also not describing their diagrams properly (no one managed to do the double diagram of collision detection, as it has two modes, properly). A couple managed to understand the idea of reusing rules (applying good CS principles) for the last part correctly, rather than trying to rewrite brand new rules.

Q5: Description of motion graphs often failed to include details of creating multi-segment (piecewise continuous) graphs and why parametric curves are used (single variable that can be mapped to time, easy to evaluate curves for both playback and rendering in a user interface). Most people mentioned slow-in/slow-out for Bezier graphs but failed to mention constant speed animation when using linear graphs. The description of arc-length parameterization was very vague (parameter value being proportional to arc length). Lack of this property means that the motion graph is evaluated at the wrong time if a linear mapping of time to parameter value is assumed. For the particle systems it was often forgotten that particle motion is calculated by specifying forces within the particle system and then applying Newton's equations of motion to calculate the motion of the particles from the forces acting on them. Most answers mentioned that particle properties were update within the particle system but didn't say how or simply used a random value for things like initial position and velocity.
The paper went down very well with the students. The marks ranged from 34/60 to 50/60, with all students but one, i.e. 8, getting 39/60 or more. The questions were well understood. The students showed a good grasp of the basics. However, answers to Q2b(iii) and Q2c(iii), which required a little more creativity than other parts, were often very disappointing, although these two parts were entirely straightforward. I cannot explain that, as students were given many exercises.

COMP 67021 High-Level Programming (Java)  Ke Chen

All students attempted questions in three sections. Their marks are roughly distributed in three intervals; i.e., six above 70%, two in the interval between 50% and 70% and one below 50%. The highest mark is 90% while the lowest one is 49%.

For short questions in Sect. A, most students have a good understanding of basic concepts and book knowledge underlying Java and general object-oriented programming in general. For comprehensive questions in Sect. C, most of students generally managed to answer questions well where the book knowledge. However, a couple of students performed poorly for questions that require some critical analysis and comprehensive knowledge in problem solving.

In summary, the overall result looks quite positive. In particular, the distribution of examination marks is well consistent with that of coursework assessment.

COMP 67321 Databases  Goran Nenadic

Q1. Most of this question was bookwork. Still, a number of students struggled with part (a), even though this has been discussed several times. Part (b) offered a practical example of update anomalies, but a number of answers did not include any specific example (e.g. changing the name of a project). Parts (c) and (d) were relatively without any major problems, although not many students discussed integration with OO languages as a potential issue for relational databases.

Q2. This question was taken by almost all students. Part (a) was typically short of providing any example of access control, and marks were lost there. The ER diagrams were mostly ok, with very few marks lost. On the other hand, a number of students had a problem to define the attributes for the EXHIBITION table.

Q3. This question was taken by only half of the students. Surprisingly, part (a) proved to be difficult for many, in particular designing the 'Head of School' table and relation 'headOf'. In many cases an ad hoc procedure was followed, resulting in an inconsistent result. Part (b) was bookwork and mainly OK. Part (c) however showed a wide range of comments, some very detailed and some too generic. A couple of answers failed to provide a 'plain English' description — providing a rather technical comment (in a loop, a command is executed...).

Q4. This question was taken by almost everybody. Part (a) was bookwork but there was a problem in describing what referential integrity was. Party (b) was the part most of the marks were gained – the only issue was with the query that required a group-by and having clauses (“Find the flight numbers and flight dates where the number of booked seats is less than 50.”). In many cases part (c) missed the main problem, with some answers discussing unrelated materials.

Q5. This question was taken by only one brave student.