Online Assessment of SQL Query Formulation Skills

Julia Coleman Prior

Department of Software Engineering
Faculty of Information Technology
University of Technology, Sydney
Sydney, Australia

julia@it.uts.edu.au

Abstract

Being able to formulate useful SQL queries is a fundamental skill required by many software development professionals. Mastering this skill is a difficult process, requiring considerable practice and effort on the part of the student. In addition, assessment of SQL query formulation skills is a process that does not appear to have been thoroughly researched, and numerous problems are inherent to the approaches commonly taken in universities to do this assessment. This paper examines two of these approaches, identifies problems with them, and then proposes another method of assessment. The way that students are assessed in a subject has a significant impact on their learning approach, and it is crucial that assessment tasks are carefully designed and implemented to inculcate a deep learning experience. The online assessment method proposed is described, evaluated, and the challenges and benefits of using it are discussed. This is a work-in-progress and the conclusion states that further validation is needed, and there is opportunity for additional research and development in this particular area of assessment.

Keywords: assessment, databases, SQL, learning approaches, online assessment, SQL query formulation, testing.

1 Introduction

Constructing database queries in Structured Query Language (SQL) is a pivotal skill required by many software developers as it underpins a range of software in a variety of domains, and is used to manipulate data and retrieve information. For example, many banking systems use databases to store their customer and account data and embedded SQL to process transactions and update balances. Retailers store inventory data in databases, and use SQL in one form or another to process sales transactions and update the inventory. Online stores that accept orders from customers over the Internet use large databases to store customer, order and inventory data, and use SQL embedded in scripts written in languages such as PHP or ASP to display data and information on their web pages. As well as using SQL for these operational activities, these systems would also make extensive use of SQL to summarise data stored in their databases to provide information necessary for decision-making at tactical and strategic levels.

Usually only one SQL statement is necessary to get a useful result, and these statements are very short (relative to the complete computer program that would be necessary to fulfil a similar purpose using a conventional programming language). Thus, there is the perception that it must be relatively simple to learn to write SQL queries. It is actually a challenging skill and 'students have many difficulties learning it' (Mitrovic, 1998). Mapping from a problem statement describing what information is required from the database into an appropriate SQL statement is not easy, as when one SQL statement is executed the database software performs numerous operations that are imperceptible to the programmer.

It is particularly difficult if one cannot see the result set that would be returned from the database when the query is executed. Yet this is how students are often expected to construct SQL queries when they are being assessed in this skill. The way that a professional software developer usually creates SQL queries is as follows. They design an initial query by deciding:

a) what database tables hold the required data,

b) how these tables need to be combined or joined,

c) what selection criteria the data needs to meet, and

d) whether the rows need to be processed individually or as groups.

Professionals verify the results of this preliminary query once it has been executed online, and if it does not accurately return the required information, they refine the SQL query and re-execute it, repeating the verification and refinement steps until they are satisfied that the query is returning the desired results.

This paper examines the difficulty of assessing students' skills in formulating SQL queries in a fair and effective manner. The first section considers this difficulty, discusses two common approaches to this assessment task and the problems therein. The second section describes a work in progress that is a possible solution to the SQL assessment problem, and the third and fourth sections discuss the challenges and benefits of this solution. The conclusion summarises the results of implementing such a solution for the first time, and suggests that further research and development needs to be done in this area.

2 Statement of the Problem

One assessment approach in a database subject that includes SQL query construction is to give students a set

of problems (descriptions of information that needs to be retrieved from the database) and to ask them to construct SQL queries as the solutions. This may be in the form of an assignment to be submitted or as a supervised, written test. This is the assessment route taken by numerous universities in their introductory database subjects (which include SQL) (Grundy 2001, Maciaszek 2001, Paradis & Barbour 2002, Webster 2002).

In my department, for instance, students studying 'Database Fundamentals' subjects have previously been assessed on their SQL skills in this manner. After working on it for a few weeks, they submitted a formal assignment in a written or typed text document. Academic staff then manually marked these assignments.

There are a number of problems with this approach. As the assignment is submitted as a written document, it is not a motivation for students to practice their query construction skills online, which is how they will use these skills professionally. The structure and evaluation of the assignment leads students to believe that the work required to master the subject is simply to write queries out manually, without the necessity of executing and verifying them against a database. The type of assessment that they will experience significantly influences students' learning strategies for a particular subject (Ramsden 1992, Biggs 1999).

The problem here is that students were passing the subject, but they did not necessarily have the requisite SQL skills. Numerous students over the last few semesters gained very high grades for the SQL assignment submitted, but obtained very poor grades on the SQL section of the exam paper. It would appear that the grades awarded to students for the assignment are not a reliable indication of their SQL skills. These high marks inflated the students' final grades, with the result that even some of the students who did badly in the SQL section of the exam passed the subject.

Constructing SQL queries is a practical skill, and cannot be gained without significant effort and repeated online practice. Most students do not put in this effort (after all, they are not assessed in this way), and consequently often find the assignment questions very difficult to answer. The paradoxically high results gained for the SQL assignment and poor results for the exam, particularly the SQL section, probably means that for some of them the easiest route is to plagiarise, at least to some extent, a willing classmate's assignment answers.

Even students who have conscientiously practiced writing out queries will not develop their skills in a useful, long-term manner. One of the difficulties for a student is conceptualising and visualising the result of an executed SQL statement. Constructing queries online, executing them, visually verifying the result and, if necessary, modifying the query until it gives the correct result internalises the query formulation skill. It incorporates the idea of learning from one's mistakes. Immediate feedback is an important component in the learning loop; Mehta and Schelicht (1998) describe this as one of the advantages of their computerised assessment in large classes.

One of the obvious solutions would be to substitute the assignment with a conventional, supervised test. This is the approach used in a university where I was previously employed, lecturing database management and SQL. Although the students here had some practical classes in the computer laboratories and so were expected to practice constructing and testing SQL statements online, the assessment approach was still not satisfactory. The written test situation also imposes an artificial environment in which the student's ability is assessed. They cannot execute the statements online and immediately see the result, and so cannot effectively verify their interpretation of a problem or the query constructed as their solution. In addition, as is the case with the assignment approach described earlier, a written SQL test does little to encourage students to adopt a deeper learning approach to the subject.

3 Proposed Solution

3.1 Motivation

In our department we have introduced an online test to assess students' SQL skills in the introductory database subjects, using software that has been developed in-house specifically to address the issues raised in the first section. There appear to be several systems available that automate submission and testing of students' programs for assessment, for example, BOSS (Joy and Luck, 1998) but the author has not been able to find software for effectively assessing SQL query formulation skills.

The aims of this assessment approach are three-fold:

(i) to assess students using an approach which accurately determines their individual SQL query formulation skills;

(ii) to assess students in a manner which closely replicates the way that they will use their SQL skills in real-world software development, as described earlier, and

(iii) to encourage students to practice and develop their SQL skills online.

In our database fundamentals subject, one of the major learning outcomes is that a student is able to construct useful SQL queries. This is a practical skill, an application of knowledge, and as such the students need to practice it regularly in order to master it.

There are numerous software packages available designed specifically for teaching SQL query formulation skills, for example, WinRDBI (Dietrich, Eckert & Piscator 1997) and SQL-Tutor (Mitrovic 1998), as well as several web sites (e.g. www.sqlator.com www.sqlcourse.com) that enable students to practice formulating and executing queries and giving them immediate, individual feedback. However, students need a convincing reason to motivate them to make use of such tools, or even database software, directly.

Biggs' (1999) concept of alignment suggests that to foster a deep learning approach by students, assessment practices need to be integrated with teaching and learning

activities and the learning outcomes. The impact that the assessment method has on the student's learning approach is described by Biggs (1999) as the 'backwash effect'. The assessment method should encourage students to take a deep learning approach, not enforce a surface one.

Together with Ramsden's (1992) suggestion that the type of assessment 'shapes the curriculum' and strongly impacts the student's learning approach, it would seem that assessing a student's SQL skills online, and in a manner similar to how they will use SQL as software professionals, would encourage them to adopt the same tactic in their learning approach.

Also, Toohey (1999) states that giving students practical, professional tasks to perform for assessment has 'clear relevance' to professional education. Ramsden (1992) quotes Newble and Clarke (1985) who established that problem-based learning that closely mirrors the type of problems met in professional life is 'more likely to encourage students to adopt a deep learning approach'.

'Performance assessment' is an approach which requires that students are able to actively demonstrate that they understand and can apply the knowledge they have gained (Biggs 1999), not simply discuss it or write about it. The online SQL test would fit into this category of assessment.

3.2 AsseSQL – an online tool to test SQL query formulation skills

A description of the online test software, AsseSQL, follows. All the data about each test to be taken are stored in a database, for example, test dates, duration, total number of marks, number of questions and type of SQL query to be tested in each question; in other words, the design or structure of the test. Also stored in this database is a query pool – a selection of SQL problems and model answers (i.e. queries) that test different types of SQL statements. The structure of each test is such that although all the students in a class will do Test1, for instance, each student will be given their own unique version of Test1 when they actually take the test.

Assume that we design Test1 so that there are 5 questions in total:

- question 1 is a SELECT on one table with one WHERE clause
- question 2 is a SELECT on one table with more than one WHERE clause, joined by logical operators
- question 3 is a SELECT on one table with a GROUP BY and a HAVING clause
- question 4 is a SELECT on two tables with a natural join
- question 5 is a SELECT with a sub-query containing a simple SELECT

In the query pool, there are a number of problems that could be used for question 1. When a particular student logs on to do the test, the program chooses one of these queries for this student's question 1, and similarly for each of the other questions in the test.

Two separate databases are used for every test. The first one has been briefly described above, and is used to administer the test data - it stores test structures, possible questions and answers, each student's questions and their answers as well as their results for every test they've taken. The other database is what we will term the scenario database and this contains the tables against which the solutions (queries) to the test questions are executed. For example, there might be an Order Entry database containing Customer, Product and Order tables for Test1. The questions for a test would all require queries to be constructed for data stored in this scenario database.

The students take the test in the faculty's computer laboratories under supervision. This is to ensure that it is the students themselves who take the test. The test software is web-based, residing on the faculty's intranet. Two levels of security need to be passed before a student can begin to take their test. The student must first logon to the intranet and in order for them to actually start taking the test, a supervisor userid and password must also be entered. This userid and password are different for every test session, and are only valid for that test session. The student is thus only given these details once every student is logged on and ready to begin the test, and no student may leave the test venue until the end of the test session. The test duration is fixed and is the same for every student, but each student's starting time is only recorded once they are through both authorisation stages, and their test will be available to them for the test duration (e.g. 60 minutes) from their individual starting time. When the student's time is up, their test is locked and the student is not be able to submit any more answers.

Once the student's test is started, the first form presented to the student lists their particular set of questions for their test (see Figure 1 in Appendix). The student may answer the questions in any order that they wish. And they may attempt to answer each question as many times as they wish, until it is correct, or their test time is up.

From this first form, the student clicks on the question that they wish to answer and are shown the answer form. This displays the question again, as well as the result set (of data) that should appear when a correct answer (query) is executed. Beneath this is an empty text box into which the student types their solution i.e. a SELECT statement (Figure 2). They submit their answer and the SQL statement will be executed against a scenario database e.g. the Order Entry database.

If the submitted answer is syntactically incorrect, an error message is displayed (see Figure 3). If the statement is executable, the data grid containing the result of the student's executed answer is displayed beneath the answer text box. If these results are not the same as the model solution's, a message stating this is shown below the answer text box, the data results of the student's query is shown and the student can compare their data result

with the required one (Figure 4). In either case the student can amend their SELECT statement and re-submit. Alternatively, they can elect to go back to the first form that lists all their test questions and choose to answer another question.

The program marks the student's answer using a pattern-matching system. It compares the data set produced by the execution of the model answer to the data set that results from the execution of the student's answer. If the data sets are exactly the same, the student's answer is flagged as correct; otherwise it is flagged as an unsuccessful attempt.

If the student's answer is correct, they will be taken back to the first form again automatically. Any correctly answered questions will now have messages next to them stating this. Questions that have been attempted but are not yet correctly answered will also have a relevant message next to them (Figure 5). The student can then click on the next question that they wish to answer.

The student may logout of the test at any time, but will be able to login again and attempt any incomplete or incorrect answers until their individual test time limit is up. In the same way, if their test window is closed accidentally, they will be able to login again and continue from where they left off, providing that their test time is not up.

The students are able to practice using the SQL test software. Ramsden (1992) emphasises that an assessment task should not be threatening and states that the lecturer should do everything possible to 'lessen the anxiety raised by assessments'. A mock test was set up and the students were able to try this out as often as they wished in a non-test atmosphere before the actual, formal test near the end of the semester. The only difference between this mock test and the actual test software is that the student may take the mock test as often as they wish, whereas the actual test may only be taken once by each student. Thus, students who used the opportunity to practice with the online test software were quite comfortable with the approach at the test time, and able to focus on constructing the queries to be assessed, without having to be concerned about how the software works and how to interact with it. The mock test also gave the students further opportunity to practice their query formulation skills online. They were given a data model and description for the scenario database to be used in the test a week before the test date, so that they did not have to consider what the tables and relationships represented during the limited test time. At the beginning of the test session, the students were also given printouts of the actual data values stored in each table in the scenario database.

Other software tools are also available in the subject for the students to practice using SQL during the semester. The database management system PostgreSQL is used, and the students can interact with this directly using its command-line interface (it is Unix-based), or using Microsoft Access (a desktop database management system) as a visual interface to the PostgreSQL databases on the Unix server via Open Database Connectivity (ODBC). Secondly, a web-based program is available that is designed specifically to allow students to practice online queries in a very similar manner to the AsseSQL software. At this stage, the package does not have formal testing facilities available, which is one of the reasons that we developed our own software. This variety of software tools reflects the real software development world where professional developers are expected to be able to apply their SQL skills to numerous and varied software systems. Moreover, it gives students ample opportunity to develop their SQL skills online, using their own choice of the software available in the laboratories.

4 Issues and Challenges of Online SQL Assessment

AsseSQL was used for the first time in the Spring 2002 semester, by 442 students. This initial use has been evaluated in the context of the three aims listed in section 3.1, as well as Toohey's (1999) considerations for the selection of an assessment method (these are summarised in section 5), using student questionnaires (which had structured questions with a Likert scale of 4), focus groups, an online discussion forum and the manual remarking of a considerable number of the submitted tests. All students who had expressed written concerns about the way that the software had marked their tests had their tests remarked by an academic, who also remarked another, random selection of student tests to verify that the marking was done fairly and as expected by the software.

Several concerns about the online test were raised in the evaluation process. The one expressed most often in both written feedback and discussion was that students wanted partial marks to be given for partially correct answers. The marking of the test is binary, in other words, the student's answer is either correct or it is incorrect. If the student's answer is partly correct, no marks are allocated. There are no marks for effort. On the whole, results are recognised in industry, not effort. If a developer is required to produce a program, the client will not consider that the work is complete until the software is performing as requested, regardless of the effort and time put into its development. In the same way, if a form displaying some information is required, for example, that has an underlying SQL query retrieving this information, and the SQL statement does not even execute, the work is incomplete and incorrect. If the statement executes but returns invalid information, the work will also be regarded as invalid. In addition, a computer system will usually only accept for execution a statement that is syntactically complete and correct. If, however, a lecturer feels strongly that they would like to allocate partial marks, the students answers are stored in a database and can thus be manually marked as well.

A number of students expressed a desire to see the actual DBMS error message when their queries were syntactically incorrect, rather than the generic Syntax Error' message that is displayed by AsseSQL. Not all students agreed, particularly ones who have worked or are working with databases in industry, and they stated that DBMS error messages are frequently vague or

incomprehensible and therefore not very useful. Nevertheless, the next version of the software will incorporate this change, as in some cases more comprehensive messages may assist students correct their mistakes.

'Different questions for different students' was an issue for a couple of students – they believed that it was 'unfair' that all students did not take the same tests, with exactly the same question set. Analysis of the discussion and written feedback indicates, however, that these students did not fully comprehend the way that the tests are structured i.e. that all question 1's, for example, test the same type of problem and are of the same level of difficulty.

Some students regarded the time limit on the test to be too restrictive – this was certainly true for students who were struggling to formulate correct answers. However, students who did well i.e. received full marks (almost one third of the students who sat the test), mostly finished in half the allocated time or less.

A rather serious problem occurred during the running of the first 2 test sessions (there were 12 altogether), and this was that the server response time was exceptionally slow for some students, and this certainly added to their stress and negatively impacted their performance. The lecturer of these students moderated the marking of these tests to counteract the effect somewhat. The online tests have been scheduled for much earlier in the semester next time around, so that the load on the server with users and jobs other than the online test is reduced, and the database server will be moved to a different physical machine to the normal faculty server used by students.

Culwin (1998) suggests that with online assessment an alternative paper-based assessment should be provisionally scheduled for a later date, as well as the formal online assessment date, as it is simply not possible to assure '100% certainty of quality of service' using a network and server(s). If it then proved impossible to deliver the online assessment satisfactorily, use could be made of the planned alternative without undue inconvenience to the students.

A concern that has been raised is that students can design a contrived solution for a particular question that is only valid for specific results i.e. those shown in the sample output. For example, where the student is required to code a reasonably complicated query that returns only a few rows and columns, one of which has the values '1, 5, 7, 9', the student could instead simply write:

```
SELECT <appropriate columns> FROM <table>
  WHERE <column> IN (1,5,7,9)
```

For this initial use, these types of queries were searched for both manually by the moderating marker and using database searches. One possible way of dealing with this problem is (during the test) to execute the model answer query and each student's answer against a second scenario database with data slightly different to that shown in the sample tables and output, as well as executing them against the database given to the students. The type of modifications to the data would be, for instance, different minimum and maximum values for groups of rows and changing the number of rows in one table related to one row in another table. An exception report could also be run in batch after the test to record student answers that differ significantly from the model answers, and the marking of these could be moderated.

Possibly the area that requires the most attention and care in this approach is the setting of the test questions. It is crucial that the problem statements are precise and unambiguous, so that students are certain which information should be retrieved from the database. Although the displayed 'model answer' results help clarify this, it is nonetheless important to have several staff members work through and try to answer the questions without seeing the model answer to check for any imprecision or ambiguity.

Whilst this may not reflect the real development world, where a client's ambiguous requirements can be clarified and verified, in a test situation, students do not necessarily have the same opportunities to check that their interpretations are valid.

5 Benefits of Online SQL Assessment

In a list of factors that should be considered when selecting an assessment method, Toohey (1999) includes:

(i) the validity of the assessment, which is how accurately the assessment reflects the learning objectives for the subject;

(ii) the reliability of the assessment, in other words that the same work submitted for assessment on different occasions should return similar results;

(iii) how well the assessment leads to and enables real learning: as mentioned by other authors (Ramsden 1992 and Biggs 1999) the way that work is assessed has an enormous influence on the approach that a student takes to learning in a subject.

As mentioned previously, the evaluation and feedback was examined to some extent fromthe perspective of the 3 aims detailed in section 3.1 as well as Toohey's factors given above.

Our first, basic aim of accurately determining individual students' query skills is achieved by doing the assessment as a supervised test, and recording and assessing individual's attempts to formulate SQL queries in a realistic situation i.e. online.

In the focus group discussions and student questionnaire responses, most students agreed that the first of Toohey's factors above – that the assessment task accurately reflects the learning outcomes - was fulfilled by the online test. Aligning the assessment task with the learning outcomes for the SQL part of the subject was one of the major motivations for introducing the online test. In order to pass the test, students had to be able to construct valid SQL queries.

One of the advantages of using a computer to perform tasks is that it is consistent, and ideally suited to doing the same tasks over and over without the repetition adversely affecting its performance as it does with humans. When manually marking hundreds of students' answers it is extremely difficult, and in fact very unlikely, that an academic staff member will be able to mark students' answers completely consistently and fairly. With AsseSQL, answers producing the same results will always be marked reliably and accurately and it thus complies with the second of Toohey's factors, the assessment's reliability.

In the formal questionnaires, the focus group discussions and informal, open-ended feedback, numerous students concurred that anticipating the online assessment task influenced the way that they went about learning and developing SQL query skills. One of the statements in the questionnaire was: 'The online test motivated me to practice and develop the skills I needed to formulate SQL queries more than I would have if I had submitted a written assignment' and the majority agreed with this statement. One student in a focus group commented that it (the online test) forced him to develop SQL skills in a way that a written test would not necessarily do, partly because of the practice software but partly because it was a more realistic approach and therefore more interesting. Other remarks included 'the online test pushed me to practice online as often as possible' and 'it [the online test] is really a good way to motivate students to learn SQL'. Clearly, this assessment approach fulfils the 3rd aim of encouraging students to practice and develop their SQL skills online (section 3.1), as well as Toohey's 3rd factor referring to the assessment's impact on real learning.

The evaluation process indicated that students consider that the assessment closely replicates the way that they will use their SQL skills in real-world software development, fulfilling our second aim (section 3.1). Some of the students taking the subject this semester were concurrently doing a semester of industrial practice with software development companies, which is a required part of their degree course. Significantly, this group of students were extremely positive about the use of AsseSQL for SQL skills assessment, particularly in the context of this second aim.

Having considered the online SQL assessment method in the light of Toohey's factors and the three aims stated earlier, there are several other advantages to this approach to bear in mind, both for the student and for the subject's academic staff.

For the students, they are able to:

- view the required data result, which is closer to what they will encounter in real system development requirements, and will help them to envisage and construct the relevant query;
- answer the questions in a manner that is similar to the way that they will construct queries when developing actual database query systems as professionals;
- verify their solution for each question by executing the SELECT statement and comparing their answer with the required results (data set). This will guide them to what amendments they need to make to their answer before re-submitting it - the majority of students stated that being able to verify a submitted answer against the sample data helped them to reach a correct solution to a question;
- receive immediate feedback on the validity of their solution – as well as getting another chance to submit an answer. This expands the learning experience so that the assessment becomes a significant part of it, not simply a means of receiving a grade, an important pedagogical aim (Ramsden 1992);
- receive consistent and fair marking.

For the lecturers, they:

- have reduced marking – as the software evaluates all the answers submitted, academic staff do not have to manually mark the tests. In our department, we also have very large classes, so the marking load on staff is considerable. It is time-consuming to mark SQL queries, and difficult to mark absolutely consistently, particularly when students' answers i.e. SELECT statements for complex problems are different from the model answers. It is hard to verify that a complex query will give the required information without actually executing it;
- can re-use some of the queries for different tests, saving test-setting time. After every test the testing space grows automatically – more queries are added for each test, perhaps more databases, and this increases the 'randomness' of future customised tests, as well as growing the pool of query problems;
- have 'ready-made' electronic records of every student's individual tests – questions, answers and marks;
- can retrieve statistics on several aspects of the tests, for instance how many attempts students made for each question, which could give an indication of which type of queries students struggle with. Ramsden (1992) emphasises that one of the functions of assessment should be that we use it as feedback to improve our teaching approaches.

Whilst limiting plagiarism for its own sake is not the prime aim of the test, it is an added bonus. It would be extremely difficult for any student to share questions and/or answers with another student. Each student takes their own customised test – even two students sitting next to one another in the lab during the test are unlikely to have the same problems to solve for each question, and will have a different 'mix' of questions. It will be virtually impossible for any student to help another student sitting any distance away from them. The latter is also unlikely in a situation where each student will probably be more focussed on trying to get the answers correctly constructed online in a limited time.

6 Conclusion

It may be that in future we will expand the software so that instead of being limited to assessment, it will also allow for unlimited SQL query construction practice and contribute more directly to the students' learning experience. We may have to set up more explicit guidelines as to how test questions should be set fairly for use in AsseSQL.

There appears to be a dearth of literature discussing effective SQL skills assessment, and this indicates an area of more formal research possibilities. Further validation of the assessment approach described in this paper also needs to be made. When the subject is next offered, we plan to implement the test earlier in the semester, and perhaps more than once, which will give us more opportunity for formal data collection and analysis on its effectiveness.

Nevertheless, evaluation of its first use indicates that the assessment is valid, reliable and adheres to the principle given by Housego and Freeman (2000), who state that technology-supported teaching is effective when based on teaching practices which motivate students to adopt a deep learning approach, not because information technology is used simply for its own sake.

Implementing a radically different approach carries a certain amount of risk. However, the initial outcomes of this endeavour indicate a useful assessment tool that has a positive effect on the student learning experience.

7 References

BIGGS, J. (1999): *Teaching for quality learning at University.* Buckingham, Open University Press.

CULWIN, F. (1998): Web hosted assessment – possibilities and policy. In *ITiCSE'98* Dublin, Ireland, p55-58.

DIETRICH, S.W., ECKERT, E. & PISCATOR, K. (1997): WinRDBI: a windows-based relational database educational tool. In *SIGCSE'97* p126-130.

GRUNDY, F. (2001): module description for COMP-207 : Databases, Department of Computer Science, Keele University, United Kingdom. Available online at: http://www.keele.ac.uk/depts/cs/modules/0102/level2/comp207.html [accessed 27/09/02].

HOUSEGO, S. and FREEMAN, M. (2000): Case studies: integrating the use of web-based learning systems into student learning. *Australian Journal of Educational Technology*. 16(3): 258-282.

JOY, M. and LUCK, M. (1998): Effective electronic marking for on-line assessment. In *ITiCSE'98* Dublin, Ireland, p134-138.

MACIASZEK, L. (2001): Study guide for INFO603 Database Systems, School of Business and Informatics, Australian Catholic University, Sydney, Australia. Available online at: http://www.comp.mq.edu.au/~leszek/uni_courses/603_01StudyGuide.pdf [accessed 27/09/02].

MEHTA, S.I. and SCHLECHT, N.W. (1998): Computerized assessment technique for large classes. *Journal of Engineering Education*. April 1998: 167-172.

MITROVIC, A. (1998): Learning SQL with a computerized tutor. In *SIGCSE'98* p307-311.

PARADIS, F. and BARBOUR, R. (2002): Course outline for 0657.219B Database Practice and Experience. Department of Computer Science, University of Waikato, Hamilton, New Zealand. Available online at http://www.cs.waikato.ac.nz/Teaching/0657.219A/course_outline.html [accessed 27/09/02].

RAMSDEN, P. (1992): *Learning to teach in higher education.* London, Routledge.

TOOHEY, S. (1999): *Designing courses for Higher Education.* Buckingham, Open University Press.

WEBSTER, M. (2002): Course outline for ZXX-4506 Introduction to Databases with Oracle and SQL, Dept of Information Services, University of Bangor, Wales. Available online at: http://www.bangor.ac.uk/is/teaching/postgrad/postgrad.shtml [accessed 27/09/02].

Appendix – Screenshots of AsseSQL

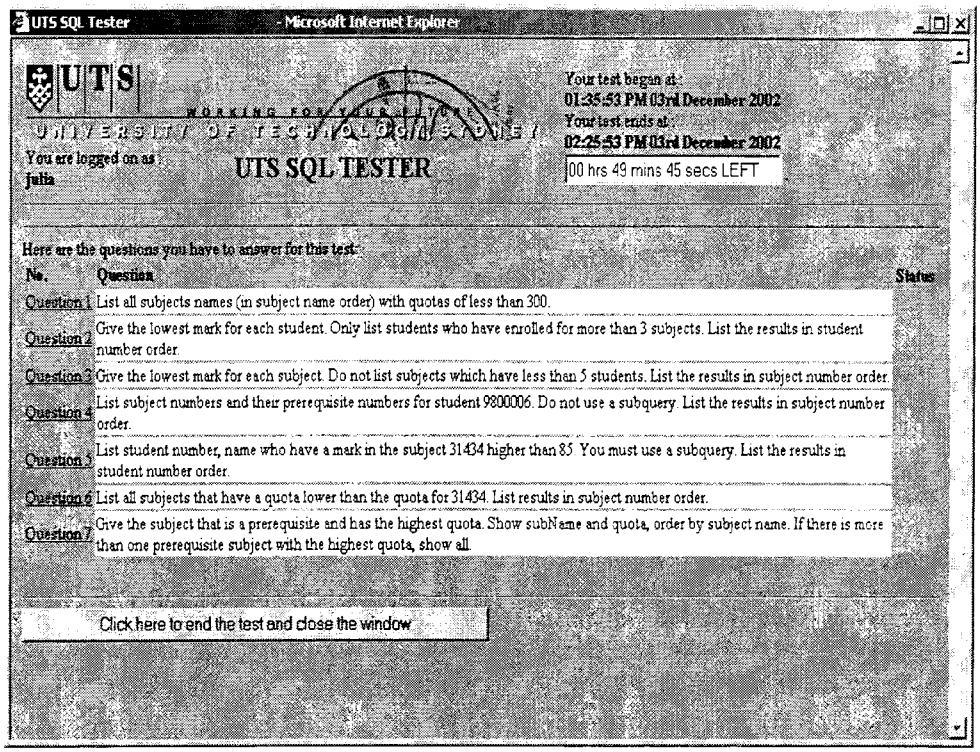

Figure 1. AsseSQL Opening Form

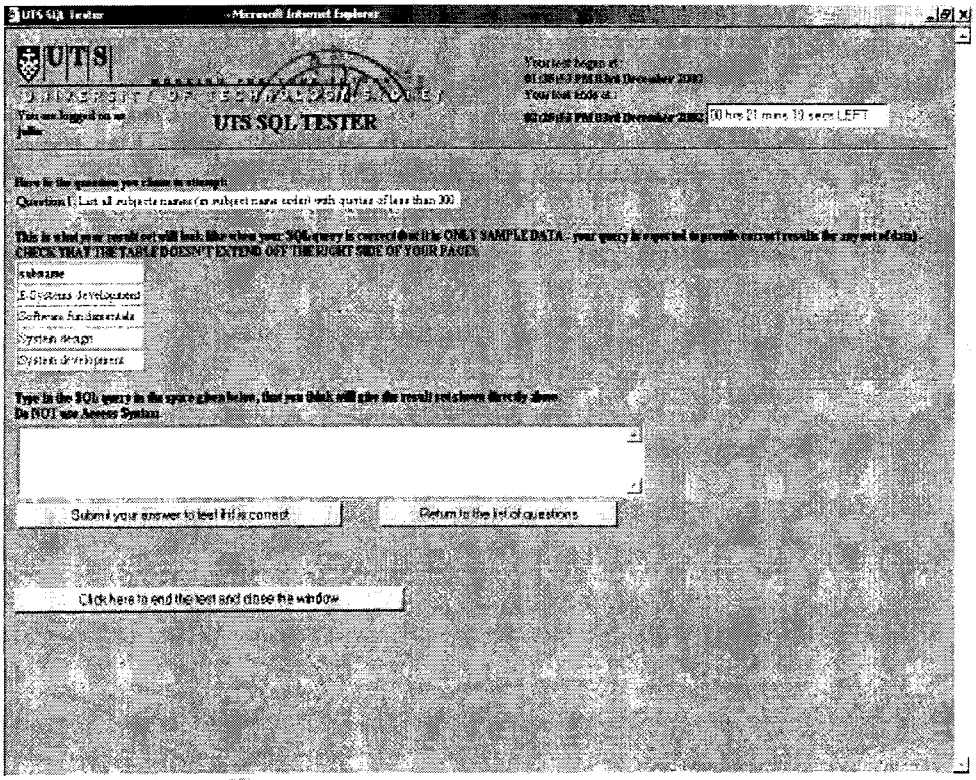

Figure 2. Answer Form

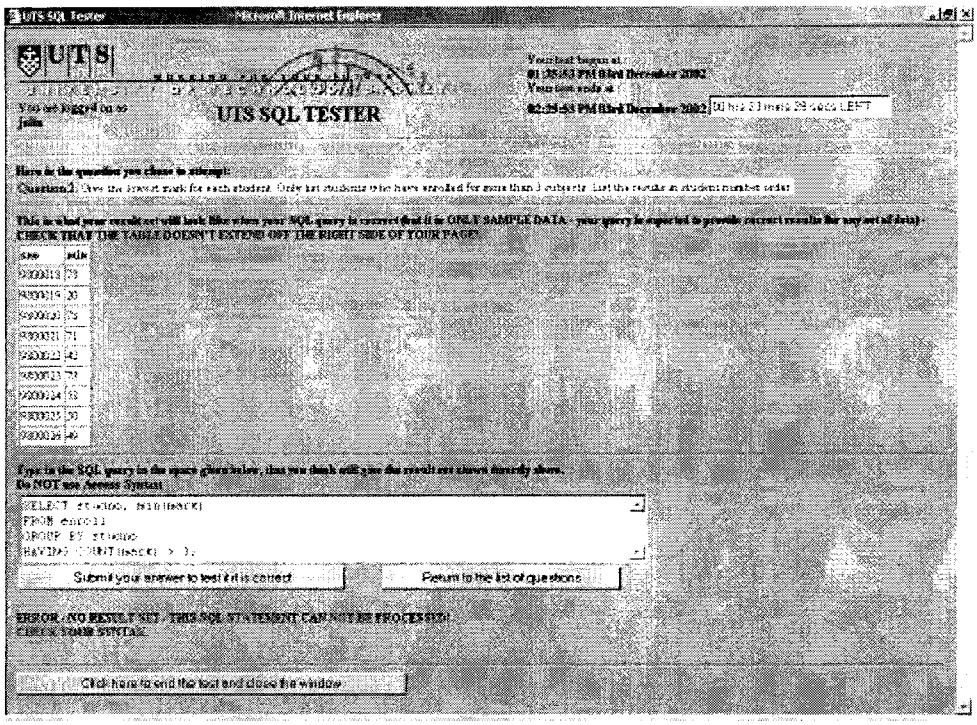

Figure 3. Error Message Displayed (attribute name misspelt)

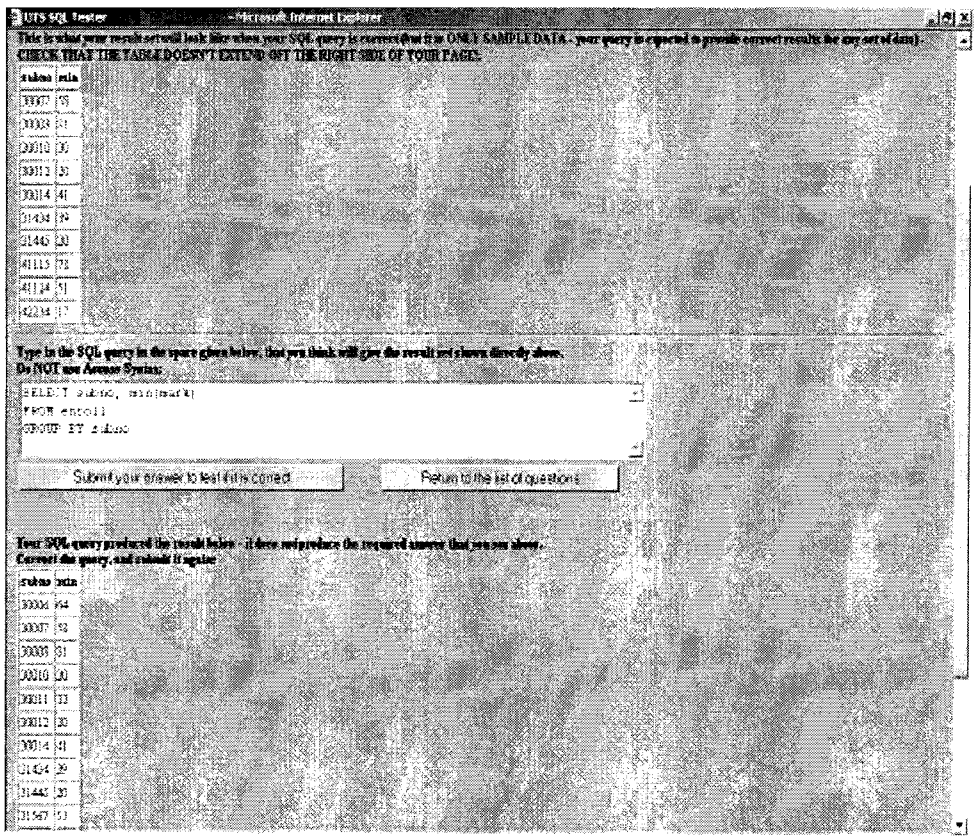

Figure 4. Query Executed But Results Incorrect (only lower part of screen shown)

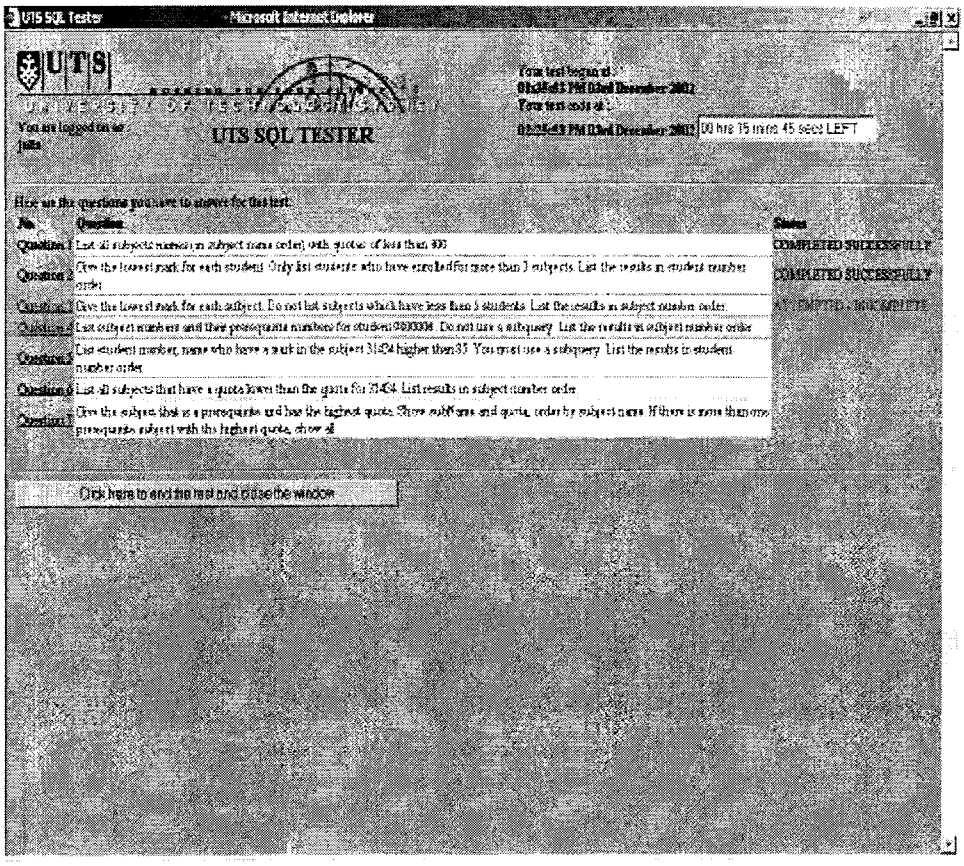

Figure 5. Main Form Showing Results of Attempted Questions