LEARN-SQL:Automatic Assessment of SQL Based on IMS QTI Specification

Alberto Abelló[1], M. Elena Rodríguez[2], Toni Urpí[1], Xavier Burgués[1],
M. José Casany[1], Carme Martín[1], Carme Quer[1]
Universitat Politècnica de Catalunya[1], Universitat Oberta de Catalunya[2]
{aabello,urpi,diafebus,mjcasany,martin,cquer}[1]@lsi.upc.edu, mrodriguezgo@uoc.edu[2]

Abstract

In this paper we present LEARN-SQL, a system conforming to the IMS QTI specification that allows on-line learning and assessment of students on SQL skills in an automatic, interactive, informative, scalable and extensible manner.

1. Introduction

In this paper we present LEARN-SQL (Learning Environment for Automatic Rating of Notions of SQL), a system conforming to the IMS QTI [3] that allows the learning and assessment of students on SQL skills in an automatic and efficient manner. SQL (Structured Query Language) is the dominant database (DB) language today, comprising commands to define schema structures (i.e. tables, views, indexes, etc.) as well as statements to manipulate data (i.e. queries, modification statements, procedures, triggers, etc.). The model underlying SQL is the relational model.

The development of software architectures that deal with automatic code correction is not new in computer science education. Several efforts have been made for automatic correction of programming problems (see [2] for a detailed discussion). In the case of the DB field, additional difficulties arise: firstly, the variety and diversity of SQL question types implies to cope with a broad set of evaluation methods to guarantee the correctness of student solutions and secondly, the added problems that a specialized and complex technology, as is the case of a database management system (DBMS), imposes.

Some tools have been proposed for students learning and assessment on SQL skills. [1, 7, 8, 9] are good examples of such tools. On the other hand, [4, 5] are automated tutor systems for SQL abilities training; therefore, the focus is on providing personalized training and guidance to students. The main drawback of all previous works is they only cover SELECT statements. They do not take into account learning technologies specifications either.

The main problem, when trying to automatically correct SQL questions, is that the solution, in the general case, is not unique. Moreover the number of correct solutions, for a given SQL question, grows rapidly as its complexity does. For example, the SQL statements: a) UPDATE DEPARTMENTS SET BUDGET=0.1*BUDGET WHERE #DEPT NOT IN (SELECT #DEPT FROM EMPLOYEES) and b) UPDATE DEPARTAMENTS D SET D.BUDGET=D.BUDGET-D.BUDGET*0.9 WHERE NOT EXISTS (SELECT * FROM EMPLOYEES E WHERE E.#DEPT=D.#DEPT), are semantically equivalent (both sentences decrease in 90% the budget of departments without assigned employees), although they are different from a syntactic point of view. Therefore an automatic correction based on string comparison between the student solution and all existing valid solutions provided by the teacher is neither feasible nor efficient.

So, we need to implement a strategy that objectively allows evaluating the correctness of the solution given by a student. This strategy depends on the kind of SQL question posed (e.g. SELECT or UPDATE SQL questions will have different correction strategies). In the case of the previous UPDATE statement, the associated strategy, as first step, needs to apply to the student solution a set of experiments representing different DB states for the DEPARTMENT and EMPLOYEES tables. Each experiment verifies a possible mistake made by the student when the number of updated rows (which is the output of the operation) in the DEPARTMENTS table coincides with the number of departments without employees. However this is not enough; as second step, for each experiment, it is also required to confirm that the departments having zero employees (and only those departments and only the BUDGET column) have been properly updated in the DEPARTMENTS table.

2. System architecture

IMS QTI deals with questions (i.e. assessmentItems) and tests (i.e. assessmentTests). More specifically, it proposes a software architecture consisting of a repository (i.e. itemBank) managed by the itemBankManager that stores the assesmentItems (which will be SQL questions in our case) that can be included and reused in different assesmentTests in a given learningSystem. There is also an authoringTool for the authors to manage assessmentItems and a testConstructionTool for the testConstructors to build assessmentTests. The tutors configure the materials in the learningSystem for the candidates, who can answer assessmentTest through an assessmentDeliverySystem, maybe under the vigilance of a proctor.

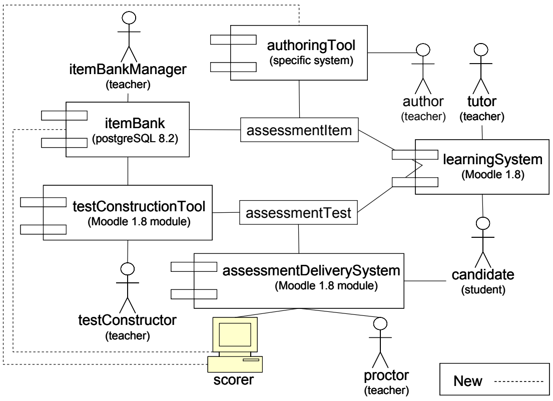

Figure 1. Extended IMS QTI Architecture.

There is one more element in this architecture: the scorer. In [3], it is defined as a person or external system responsible for assessing the candidate's responses. Thus, since one of our goals is to automatice the assessment process, we choose it to be a system as emphasized in figure 1. We have implemented it as a Web Service (WS) which is in charge of SQL questions assessing. A WS is a software resource available in the Web at the disposal of anyone who needs it. This kind of construct improves by itself the flexibility in the implementation and geographical distribution of the components. In our case, we needed to extend the IMS QTI architecture, because the assessmentDeliverySystem asks the scorers to assess candidate responses, but also the authoringTool calls them to generate the correct output for the battery of experiments of an assessmentItem. In turn, a scorer needs to query the itemBank in order to get this same battery of experiments (to retrieve the defined experiments to be evaluated, as well as to retrieve and store the correct outputs of each of the experiments).

3. Conclusions

The design of LEARN-SQL has been driven by the requirements of automatization, interactivity, informativeness, scalability and extensibility.

From students' perspective, LEARN-SQL has proved its usefulness since students can access the tool anytime and anywhere obtaining automatically semantic feedback and grading, enhancing their learning in SQL skills. The response time we have obtained during the evaluation (in a group of 40 students) is about 1-2 seconds for query and data modification items, and 5-6 seconds for schema modification and optimization items.

From teachers' perspective, LEARN-SQL helps them to teach SQL skills while enforcing consistency in grading and helping to considerably reduce their marking load, although manual qualitative assessment of student responses is not excluded. We would like to point out that our architecture allows not only the reuse of the SQL assessment items in different tests, but also among different subjects.

4. Acknowledgments

This work has been supported by the Improvement of Learning Quality Program (2007MQD00202 project) funded by the Catalan Government and by the PERSONAL research project (TIN2006-15107-C02) funded by the Spanish Ministry of Education.

5. References

[1] Dekeyser, S., de Raadt, M.., Lee, T.Y., "Computer Assisted Assessment of SQL query Skills", *ADC,* 2007.

[2] Douce, C., Livingstone, D., Orwell, J., "Automatic Test Base Assessment of Programming: a Review", *ACM Journal of Educational Resource in Computing*, 5(3), 2005.

[3] IMS Learning Consortium, "IMS QTI Specification", available at: http://www.imsglobal.org.

[4] Kenny, C., Pahl, C., "Automated tutoring for a database skills training environment", *SIGCSE'05*, 2005, pp. 59–62.

[5] Mitrovic, A., "Learning SQL with a computerized Tutor", *Proc. of SIGCSE'98*, 1998, pp. 307–311.

[6] Polsani, P.R., "Use and Abuse of Reusable Learning Object", *Journal of Digital Information,* 3(4), 2004.

[7] Prior, J., Lister, R., "The backwash effect on SQL skills grading", *Proc. of ITiCSE'04*, 2004, pp. 32–36.

[8] Sadiq, S., Orlowska, M.., Sadiq, W., Lin, J., "SQLator: an Online SQL Learning Workbench", *Proc. of ITiCSE'04*, 2004, pp. 223–227.

[9] Soler, J., Prados, F., Boada, I., Poch,, J., "Utilización de una plataforma de e-learning en la docencia de Bases de Datos", *Proc. of JENUI,* 2006, pp. 581-588.