Support of Database Skills Testing

Henrieta Telepovska
Technical University of Kosice, Department of Computers and Informatics, Letna 9, 042 00 Kosice, Slovakia
Henrieta.Telepovska@tuke.sk

Abstract

The paper deals with a database system (DBS) component which is used for support of database skills testing of students. Testing is focused on Structured Query Language, especially statements of Data Manipulation Language, and relational algebra. DBS component consists of subcomponents which are realized as plugins to Learning Management System (LMS) Moodle. DBS component also enforces designed security policy and supports of errors processing.

1. Introduction

At present, support of training by various Learning Management Systems (LMS) is normally part of education at many universities ([2][3][4][5][6][16]). LMS Moodle is used at Department of Computers and Informatics, at Technical University in Kosice [17]. Our LMS Moodle consists of over 60 courses in all three levels of higher education. LMS provides following tools from the perspective of teaching:

- Evidence of users course
- Creation of study materials
- Acceptance and evaluation of homeworks, students projects
- Quiz module as tool of test creation
  And next useful tools ([15]).
  Extending the functionality of LMS Moodle is using one of a number of API modules offered by the Moodle system for example:
- Activity modules
- Administrators reports
- Authentication modules
- Course formats
- Course reports
- Registration modules
- Test reports
- Import/export of questions
- Question types - very important for teachers and for teachers' intentions.

Database Systems course contains learning materials focused on the theory of database systems and Structured Query Language (SQL). The whole course is divided into thematic parts and each thematic part has two units – theory and practice. Each exercise includes solved practical examples and examples intended as homework.

Quiz module is very important component of LMS Moodle. It is intended for testing and verification of students' knowledge. Moodle offers standard types of questions which are possible to be used for tests creation. It follows list of standard types of questions ([15]):

- Calculated – it is way for creation of numeric questions using variables (i.e. x, y, …) which are replaced in created test with random values.
- Description – it displays some text without requiring answer.
- Essay – a student writes his answer as an essay which will be reviewed only after reading and reviewing by teacher.
- Matching – a question may contain a short introduction, a student is presented with several questions and answers. For each question there is only one correct answer.
- Embedded Answers – questions consist of a passage of text (in Moodle format) that has various answers embedded within it, including multiple choice, short answers and numerical answers.
- Multiple Choice – a student chooses from multiple answers to answer a question.
- Short Answer - an answer to the question is a word or short phrase.
- Numerical - from the student perspective, a numerical question looks just like a short-answer question. The difference is that numerical answers are allowed to have an accepted error. This allows a fixed range of answers to be evaluated as one answer.
- Random Short-Answer Matching - may include a short introduction, the respondent is presented with several questions and several jumbled answers. For each question there is exactly one correct answer.
- True/False - there are only two possibilities, answer is either true or false.

However, these are not always sufficient opportunities, thus Moodle offers also the possibility of expanding the set of given types of questions by own types of questions. For this extension, system Moodle provides template that makes the process of development of new type question easier. It is also necessary to modify questions database schema by new type question and to create new plug-in to Moodle ([8]).
A prototype DBS component that extends LMS Moodle quiz module has been designed and implemented to verify the knowledge and practical skills of students of SQL and relational algebra.

2. Structured Query Language

Structured Query Language (SQL) in earlier versions also known as SEQUEL (Structured English-like Query Language) belongs to a group of languages that work with relational databases [1]. SQL is available practically to all hardware and software platforms and includes all aspects of creating and managing databases. The relational approach is preferred to include all required functions into one comprehensive language. Therefore, the following operations must be supported:

- Database creation – a definition of database structures on external, conceptual and internal levels must be supported. External level is related to usage of particular programs and it can be created by views. Conceptual level consists of relations, attributes and keys attributes. Database performance may be affected by the definition of indices, which should be defined on internal level.
- Data manipulation – basic operations are:
  - Retrieving records according to user specified conditions
  - Inserting records into relations
  - Modification of records
  - Deleting records
- Transactions ensuring
- Other functions for database maintenance
  - Data security and data integrity
  - Checking data definition and users in data dictionary.

SQL statements processing are based on operations of relational algebra ([1]). SQL query is transformed into sequence of relation algebra operations. SQL is a declarative high-level language interface, where the user only defines what should be the result and decision is left on the Data Base Management System (DBMS), how it is implemented. Structured Query Language consists of following parts:

- Data Definition Language (DDL) – creating, altering and dropping database objects
- Data Manipulation Language (DML) – manipulation with database objects
- Data Control Language (DCL) – management of access to database objects
- Transaction Control Language (TCL) – management of transactions

DBS component described in this article is focused to DML statements – select, insert, update, delete.

3. Relational Algebra

The relational model provides concepts and rules for defining the structure and terms of the database schema. Relational algebra is set of operations which are used to manipulation of relations. These operations are used to make queries to the database, for example selection of records from the relations, a combination of related records from several relations. Result of each operation is relation again, which can be further manipulated. Operations of relation algebra are divided into two groups:

- Set operations – union, intersection, difference and Cartesian product.
- Relational operations – selection, projection and join.

([1]).

4. Architecture of DBS Component

Figure 1, shows the location of DBS component in LMS Moodle architecture. More detailed architecture of DBS component is displayed on Figure 2.

Database instance is running on database server where exam database schema is stored. A user communicates with the LMS Moodle and also DBS component via web client.

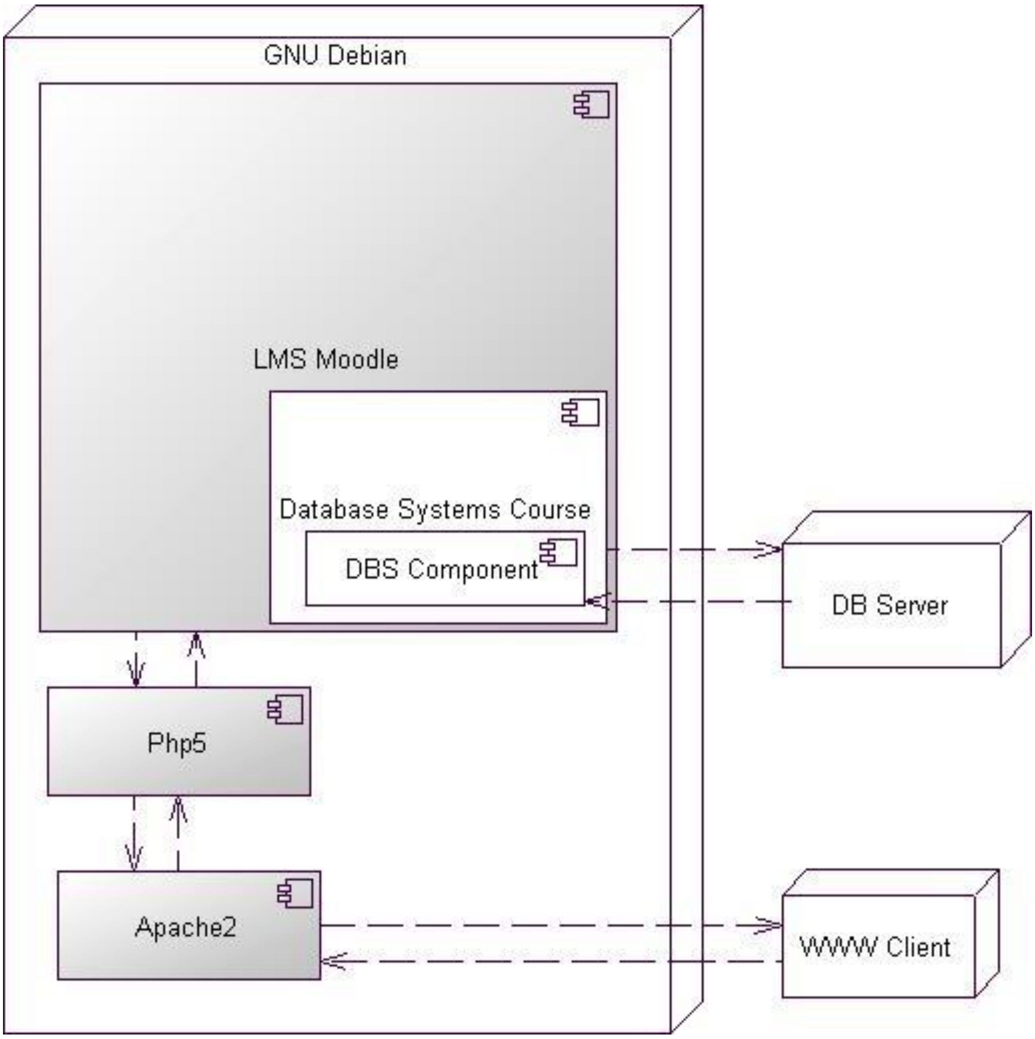

Figure 1 The location of DBS component in LMS Moodle architecture

Database systems (DBS) component consists of 6 subcomponents – 5 subcomponents for dynamic evaluation of SQL statements and one subcomponent for dynamic evaluation of query which is written by operations of relational algebra. Each subcomponent is implemented as a plug-in to Quiz module of LMS Moodle and it presents a new type of question in LMS Moodle. Php5 and Apache2 were used for subcomponents implementation.

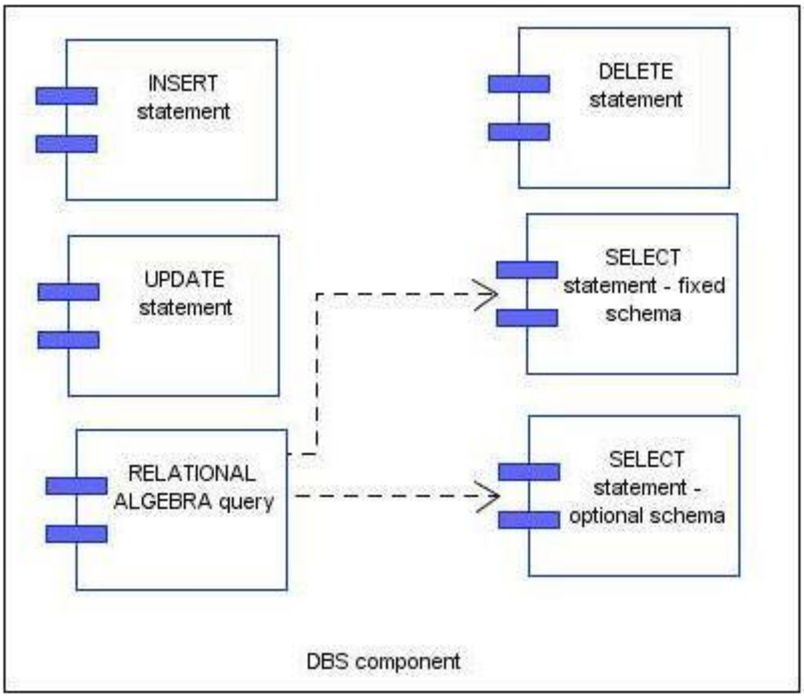

Figure 2 Architecture of DBS component

5. Subcomponents of Dynamic Evaluation SQL Statements

Subcomponents for dynamic evaluation SQL statements are:

- SELECT statement – fixed schema
- SELECT statement – optional schema
- INSERT statement
- DELETE statement
- UPDATE statement

The principle of all subcomponents is based on comparison of results of the author's questions with the student's solution. Two queries – author-teacher query and student query – are sent to the database server. Results of queries are two relations that are processed and evaluated by our plug-ins. The relations are compared. If they are equal, student's solution is correct and student obtains corresponding grades. If the relations aren't equal, student's solution is not correct and possible error is identified. Processing of insert and update statements, requires solving transactions. Delete statement is transformed to select statement and it is processed as select statement. Subcomponent SELECT statement – fixed schema – uses fixed defined database schema for evaluating solutions. Database schema is predefined (implicitly given) during question creation. Other subcomponents – INSERT, UPDATE, DELETE statement, RELATIONAL ALGEBRA query – use database schema, which is specified during question creation. Any database schema on database server can be used if it satisfies requirement of security policy.
Inputs of subcomponent – two queries (author of question, student)
Outputs of subcomponent:

- report on the correctness / incorrectness of the result
- in the case of an incorrect result - information about possible error

5.1. SELECT Statement Subcomponent

It is based on comparison of answers given by database server ([8][10][11]). This method completely omits the need to program something already programmed and uses the abilities of database server. It is not necessary to implement any syntax and semantic analysis of correctness of student's SQL statements. The whole parsing command is performed on the database server and is encapsulated in the execution of the selected SQL statement. If database server doesn't return any error, a statement is syntactically correct. If the SQL command is performed right, it remains only to compare the results of the student's solution with the results specified in the definition of the questions. If the results match, the student answered the question correctly; if not then the student was not successful.

5.2. INSERT Statement Subcomponent

Evaluation of insert statement uses database server like in other cases of dynamic queries evaluation [12]. The database server checks the syntax of the command, if it is wrong to insert module, an error is returned. If the syntax is correct, database server verifies the correctness of the statement. Insert command changes the contents of the table. After evaluating insert statement from the student and creator of question, table is returned to its original state. Processing insert statement is enclosed in a transaction; table is returned to its original state by rollback operation. Before making ROLLBACK command (after the INSERT statement) is calculated table checksums for students and authors insert command. If the two checksums are the same, both commands have changed the table in the same way, so students insert command is evaluated as correct.

5.3. UPDATE Statement Subcomponent

Like in the previous queries evaluation, database server is also used in update statement evaluation. Database server checks syntax statement. If syntax statement is incorrect, error message is returned into update subcomponent. If syntax is correct, SQL statement is verified. Processing of update statement is enclosed in a transaction. Both statements – student's and author's – change the contents of table. After evaluation of student's answer, the table is referred to the original state by rollback command. Evaluation is based on calculation of table checksums for student's and author's update statements. If both checksums are equal student's answer is correct. Otherwise it is evaluated as incorrect. ([12])

5.4. DELETE Statement Subcomponent

Delete module transforms delete statement to select command. Processing of transformed delete command is identical to select command. ([11])

5.5. RELATIONAL ALGEBRA Query Subcomponent

Relational Algebra module is intended to processing query which is written by relational algebra operations. Relational algebra query subcomponent needs entries in text format; a grammar was designed and created. Special symbols of relational algebra operations - ∏ (projection), σ (selection), ⋈ (join) – were substituted by the agreed keywords (terminal symbols of grammar). Syntax correctness of relational algebra query is checked by the parser. Based on designed grammar correct student's query is transformed to select statement that is processed by select statement subcomponent. ([9][7][10][14])

6. Support of Errors Processing

SQL statements (select, insert, update, delete) and relational algebra query are inputs to DBS component. During SQL statement evaluation, some errors can occur. These situations were analyzed and following cases were specified ([11][13]):

- syntactically and logically correct statement
- syntactically incorrect but logically correct statement
- syntactically correct but logically incorrect statement
- syntactically and logically incorrect statement
- no statement was written.

Syntactically and logically correct command is the best potential entry. It means no syntax errors occurred and also statement is logically correct. Expected results are returned and query can be finally evaluated.

Next potential entry is syntactically incorrect but logically correct statement. There was defined set of possible errors that can be repaired, for example typing error in keyword, table name, column name, missing keywords, missing delimiters in column list or at the end of the command. This statement is logically correct and it would be evaluated as correct if particular syntax error didn't occur. In these cases student's solution can be considered correct after visual check by teacher. In next version of DBS component prototype a student is able to repair his errors.

Other options are already logically incorrect. Thus, whether there is a syntax error occurred or not, the command is evaluated as incorrect. SQL query doesn't return required result. The following events can cause incorrect situation – missing some clause of statement, wrong use of join operator, incorrect select list, incorrect "where" condition, wrong use of group function and related clauses, incorrect subquery.

A situation when no query was written belongs to logically incorrect possibilities.

7. Security Strategy of DBS Component

We can say that security strategy of DBS component is realized on three levels [10]. The first level - identification and authentication - is ensured by LMS Moodle. Identification relates with student identity that is specified by his username. Student username is unique login on our university and it is assigned to a student in the first year of bachelor study. Authentication is verification of student identity by password. It is performed to the central authentication system of university.

The second level is implemented on database server side. Database schema that is designed for examination examples is stored on database server. Students don't have direct access to the schema objects during testing process. There was created a special user1, through which students are accessing the database schema objects. This special user is identified and authenticated over database server.

The third level of security strategy is authorization of the special user1. There is defined access to database objects – select, insert, update or delete operations. There was created a special user2 that is owner of exam database schema

Standard question type has its form that is typical for LMS Moodle and consists of several sections. Each subcomponent of DBS component is a new type of question. There were designed and implemented forms for each new type of question. New section was added – database connection setup. The new section contains information needed for connection to database server – user1, his password, hostname and service name. This information is confidential for student.

8. Integration and Cooperation DBS Component with Other Program Systems

Information Technologies for Knowledge Transfer system (IT4KT) provides a platform for integration and cooperation of various program systems or services. Some program systems, as it can be seen on Figure 3, for example DBS component as part of LMS Moodle, prototypes of checking correctness of database (db) schema or checking originality of database (db) schema are implemented as standalone solutions and they communicate with other systems or services via IT4KT system.

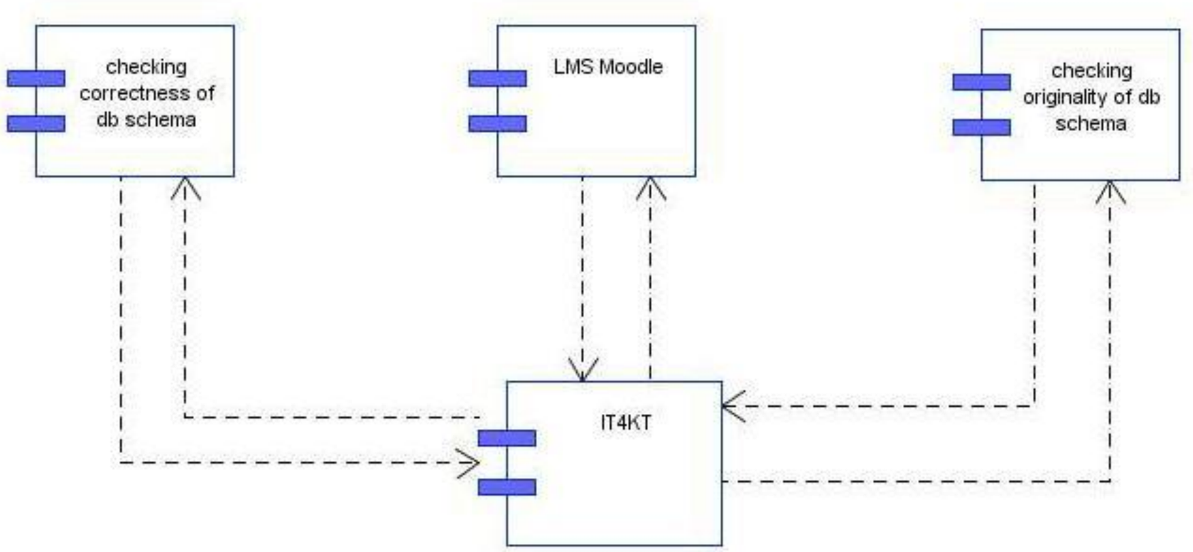
checking correctness of db schema
LMS Moodle
checking originality of db schema
IT4KT

Figure 3 Integration and cooperation with other program systems

9. Conclusion

SELECT Statement subcomponent is already in real operation for several years. It was used in teaching and also in examination. Other subcomponents are in the testing phase but they are used in teaching. Our solution is more or less dependent of LMS Moodle version and also php version. Upgrade of these parts can affect DBS component execution. At present we are modifying architecture of current DBS component. Next release of DBS component will have SOA architecture.

Acknowledgement

This work was supported by the Cultural and Educational Grant Agency of the Slovak Republic under project No. 050TUKE-4/2013: “Integration of Software Quality Processes in Software Engineering Curricula for Informatics Master Study Programme at Technical Universities – Proposal of the Structure and Realization of Selected Software Engineering Courses.”

10. References

[1] Elmasri,R., Navathe, S.B., Fundamentals of Database Systems, The Benjamin/Cummings Publishing Company, Inc., 1., 2. Editions.

[2] Raadt, M., Dekeyeser, S., Lee, T.Y., “A System Employing Peer Review and Enhanced Computer Assisted Assessment of Querying Skills”, Informatics in Education, 2007, Vol. 6, No. 1, 163–178, Institute of Mathematics and Informatics, Vilnius, http://www.mii.lt/informatics_in_education/pdf/INFE104.pdf.

[3] Sadiq, S., Orlowska, M., Sadiq, W., Lin, J., “SQLator – An Online SQL Learning Workbench”, In ITiCSE '04 Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education, Leeds, UK, pp. 223-227, ACM Digital Library, retrieved March 2010, http://portal.acm.org/citation.cfm?id=1026487.1008055.

[4] Soler, J, Boada, I., Prados, F., Poch, J., Fabregat, R., “An Automatic Correction Tool for Relational Algebra Queries”, retrieved March 2010, http://bcds.udg.edu/publications.php?year=2007.

[5] Dekeyser Stijn, de Raadt Michael, Lee Tien Yu, “Computer Assisted Assessment of SQL Query Skills”, Queensland, Australia: University of Southern Queensland, Department of Mathematics & Computing, 2007. 10 p.

[6] Prior C. Julia, “Online Assessment of SQL Query Formulation Skills”, Sydney, Australia: University of Technology, Faculty of Information Technology, 2003. 10 p.

[7] Telepovská, H., Havlice, Z., “Relational Algebra Knowledge Assessment in Practice”, Journal of Communication and Computer, Vol. 9, no. 2 (2012), p. 226-233. - ISSN 1548-7709.

[8] Telepovská, H., Tóth, M., “New Type of Question in LMS Moodle”, International Conference on Information Technology and Development of Education ITRO 2012 Proceedings: Zrenjanin. Zrenjanin: University of Novi Sad: p. 286-291. ISBN 978-86-7672-167-2.

[9] Telepovska, H., Toth, M., “Support of relational algebra knowledge assessment”, Emerging Trends in Computing, Informatics, Systems Sciences, and Engineering. - New York : Springer-Verlag, 2013 P. 475-485. - ISBN 978-1-4614-3557-0

[10] Toth, Matúš – Spracovanie relačnej algebry v systéme Moodle, Diploma thesis, Technical University in Košice, 2009

[11] Biroščák Matúš - Podpora SQL príkazov v LMS Moodle, Diploma thesis, Technical University in Košice, 2011

[12] Muľar Peter - Podpora SQL v LMS Moodle, Diploma thesis, Technical University in Košice, 2012

[13] Kaleta Robert, Podpora písania SQL príkazov, Diploma thesis, Technical University in Košice, 2013

[14] Grega Lukáš – Modul relačnej algebry v LMS Moodle, Bachelor thesis, Technical University in Košice, 2012

[15] Moodle Documentation (2010). http://docs.moodle.org

[16] Szabó, C., Model-based software testing, Testing Fundamentals : Introduction to Software Verification Theory. - Košice : TU, 2013 S. 77-140. - ISBN 978-80-553-1282-8

[17] Genči, J., Methods to ensure higher variability of knowledge tests in the moodle LMS environment, Emerging Trends in Computing, Informatics, Systems Sciences, and Engineering : Lecture Notes in Electrical Engineering 151. - New York : Springer, 2013 P. 447-453. - ISBN 978-1-4614-3557-0