ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

UDC 004.85 **doi: 10.32620/reks.2021.2.07**

A. CHUKHRAY, O. HAVRYLENKO

National Aerospace University "Kharkiv Aviation Institute", Ukraine

THE METHOD OF STUDENT'S QUERY ANALYSIS WHILE INTELLIGENT COMPUTER TUTORING IN SQL

*Recently, IT specialties have become one of the most demanded specialties in the world labor market. Simultaneously the traditional teaching in conditions of mass production, even with a professional teacher, has a significant drawback – the fundamental impossibility of adapting to each student. Since the 60s of the twentieth century, researchers worldwide have been developing various computer-tutoring tools that have, less or more, adaptive functions. Nevertheless, the task of the perfect computer tutor development is still far from being solved. The article's **research subject** is the process of student requests analyzing during intelligent computer tutoring in SQL. The **main goal** is to develop a method for analyzing the student's SQL queries. The **purposes**: to form a general scheme and features of the method for analyzing student's SQL queries based on the principles of technical diagnostics and methods of lexical, syntactic analysis of computer programs; to develop methods for parse tree construction; to create methods for comparing reference and real SQL queries according to their similarity rate; to demonstrate the function ability of the developed methods on specific examples. The **methods** used computer programs the automatic testing method, the computer programs lexical and syntactic analysis methods, the computer programs parsing trees construction methods, the objects diagnosing method based on comparison with a reference, the strings analysis methods, the method of q-grams. The following **results** were obtained: the student's SQL queries analysis method was formed based on a system approach including automatic testing on real data, building query-parsing trees, comparison with a reference, and comprehensive determination of the queries similarity rate. The **scientific novelty** is the improvement of the method for the student's SQL query analysis during intelligent computer training in SQL query composition.*

***Keywords:** intelligent computer systems; structured query language; tutoring; SQL query analysis; parsing tree.*

Introduction

In the recent decades information technology (IT) significantly transformed the environment of human life. They are currently used as a main tool for the information and knowledge processing [1, 2].

In such conditions high-quality IT training is required in the different domains. An integral part of most complex IT systems are relational databases (RDB) and algorithms for data manipulation based on the Structured Query Language SQL. However, the traditional learning of SQL, similar to many other complex domains in terms of group studying, usually does not result in the required quality training of each student. The main issue is the complexity of the adaptive approach providing by the tutor in the case of multiple students task individual solving [3]. Such approach has to take onto account difference in skills and working pace of students, as well as limitation of human tutor cognitive abilities.

According to many researchers [4-9, 12] the solution can be found in designing and implementing computer tutoring programs which have large computational power and advanced intelligent skills.

The most famous ITS for learning SQL is the SQL-Tutor system [6, 7], developed at the University of Canterbury, New Zealand. Each task in it consists of a data model and information needed to be obtained. Students write SQL queries by filling out forms. After request completion, SQT-Tutor analyzes student requests to find errors and provides multiple levels of feedback and tips. Students independently choose the level of feedback they wish to receive. After receiving feedback and editing the request, a student can again send it for consideration by the system.

For the purpose of step analysis SQL-Tutor contains many constraints, which includes two parts: a relevance condition and a constraint satisfaction condition. The reaction occurs only if the first condition is met, and the second is not. This means that the constraint is violated and ITS identifies the corresponding feedbacks.

Although the restrictions were conceived by the authors as domain-independent, many of them deal with a correct solution stored in the system. For example, one of the constraints might check the FROM section of the query to see if the value filled in by the student is equal to the value in the correct solution. The relationship between steps, learning events, and constraints in the system is fairly straightforward. Each constraint meets the competency component.

On the basis of the above principles, other ITS have been developed [10, 11].

Formulation of the problem

Despite the repeated presentation in publications by the developments as based on a fundamentally new and effective theory of constraint-based modeling (CBM) [12], in fact, this is not the case, since any restriction can be reduced to the well-known rules “if – that". A number of restrictions are created for the purpose of syntactic comparison between the only ideal solution and the student's solution. Naturally, these restrictions do not give a chance to alternative solutions that may be more rational than those created by the authors.

An alternative approach, implemented in the form of rules-models of errors, is presented in [13]. It describes 37 classes of the most common semantic mistakes made by students while SQL queries construction.

It is also worth to mention such systems as AsseSQL [14] and SQLator [15] – for assessing student requests; SQLify [16] – a tool that implements semantic feedback; and also an intelligent computer system teaching SQL from the University of Malaysia [17]. In the latter, the student is limited only by entering data into the fields SELECT, FROM, WHERE, which significantly limits the flexibility of the system.

Therefore a new development approach to ITS in SQL is required, combining existing effective techniques with the new methods of the SQL queries analysis and comparison.

To achieve the stated objective following tasks have been formulated and solved:

1. A method of student’s SQL queries analysis development.

2. Parsing and syntactic tree building methods development.

3. A method development for two SQL-queries comparison aimed to detect their similarity.

The student’s SQL-queries analysis method

Because of the queries presentation redundancy in SQL for any problem more than one correct SQL-query might exist. Hence, it is necessary to verify whether a student's SQL-query fit a certain set of standards for a concrete given task.

In the case of resemblance to one of the reference solutions system identifies the student's solution as known. Each reference query stored in the system is characterized by a set of parameters that determine its quality and, consequently, knowledge and skills of a student. Such parameters, for example, can include performance and required memory capacity, the number of characters in a query that determines the reading and understanding convenience of a request by another person.

If the system does not identify the query as known it could be executed on a real database with multiple data sets. As the result of a data manipulation query is a set of certain tuples all the correct answers must return the same set of data for the same initial data set. Hence with high probability we can conclude that if the query is not known to the sys-tem but on the series of initial data sets output the same set of tuples as the reference solution then this query can be classified as the correct answer.

ITS based upon the described approach works according to the schema presented on fig.1, where high S – high similarity, low S – low similarity, which is obtained experimentally.

After a new task for student has been generated the program displays the text with the assignments, ER-diagram of the given DB, and a field for student’s answer in the form of SQL-query.

Once the answer was confirmed system starts analysis according to the following script:

1. The system estimates similarity rate between the student’s answer and reference SQL-queries.

2. Finding similarity rate is compared with some threshold.

3. In the case of high similarity the system operates as follows:

3.1. Lexical and syntactic analysis of SQL-query carries out and a syntax tree is built.

3.2. The parsing tree is compared with the reference syntax tree, which has the highest similarity with the student’s query. Every such a model of SQL-query is described by a set of parameters that characterize the quality in several aspects: performance and memory capacity requirements, readability.

3.3. If the system found a mismatch of some nodes of the trees it informs about the error location.

3.4. If the parsing trees are identic the student’s answer is considered as correct, and a new task is generated.

4. In the case of low similarity between the student's query and reference queries the system runs both queries on the real database and then compares resulted sets of tuples.

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

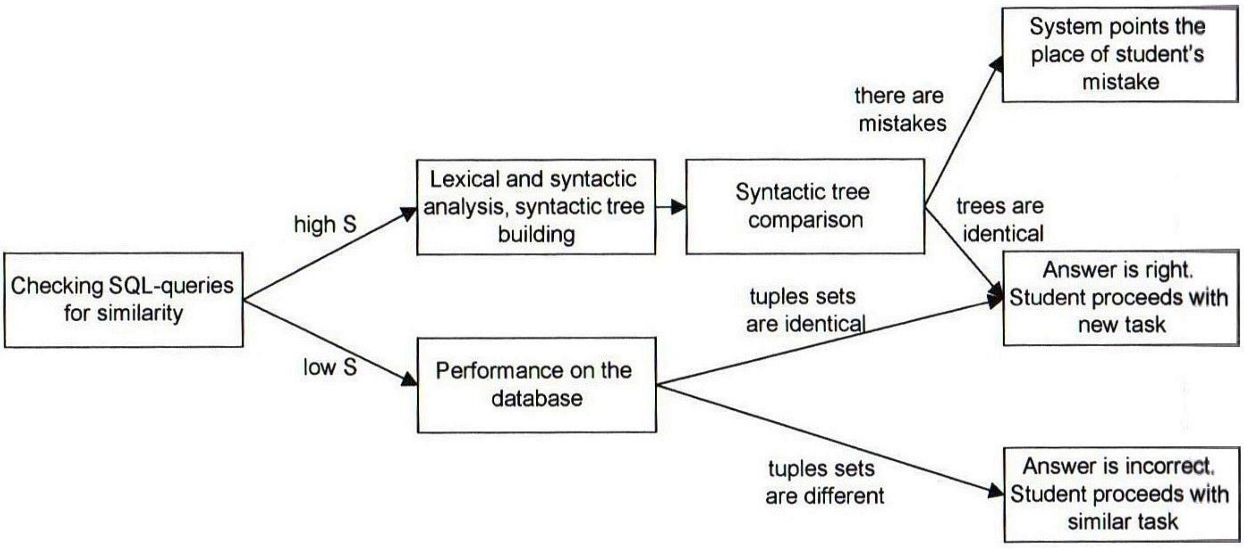

Fig. 1. The schema of the student's SQL-queries analysis method

4.1. If the sets of tuples match exactly one with another the system considers the student's answer as correct, and move to the next task.

4.2. If the sets of tuples differ, the answer is considered as wrong, and a student has to solve another task from the same topic.

The SQL-queries parsing method

SQL is a language which can be described in Backus-Naur Form (BNF). Its grammar is context-free (CF grammar). It allows developing SQL compiler in this project utilizing the programs LEX/YACC.

The main hypothesis of this paper is that the pedagogically correct feedback can be given by comparing syntax trees of the reference and the student SQL-queries.

Fig. 2, 3 show examples of the parsing trees for the reference and student queries. After the trees traversal and nodes comparison the mismatch at one of the nodes has been detected.

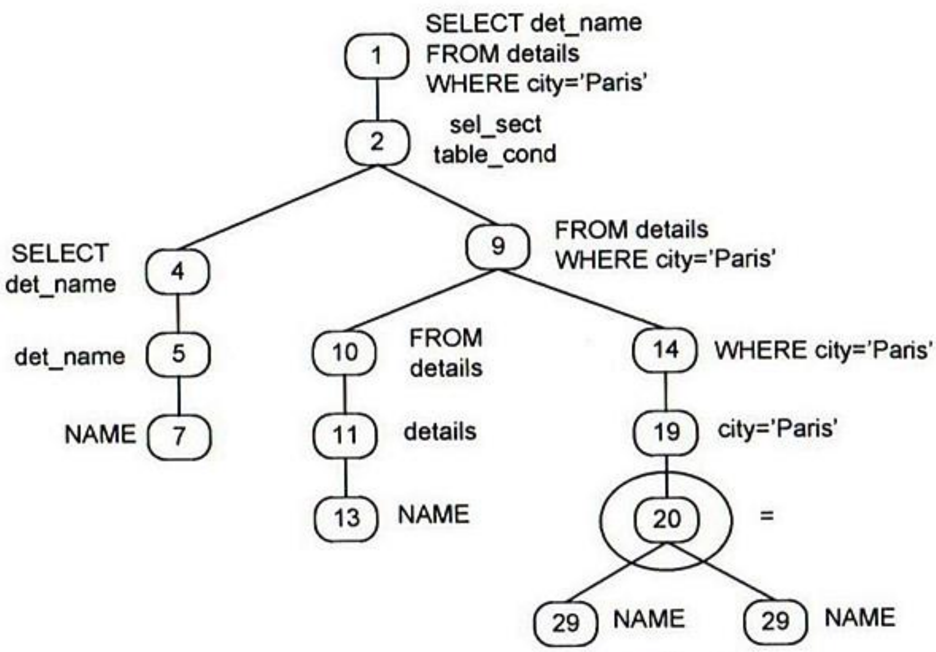

Fig. 2. The reference SQL-query parsing tree

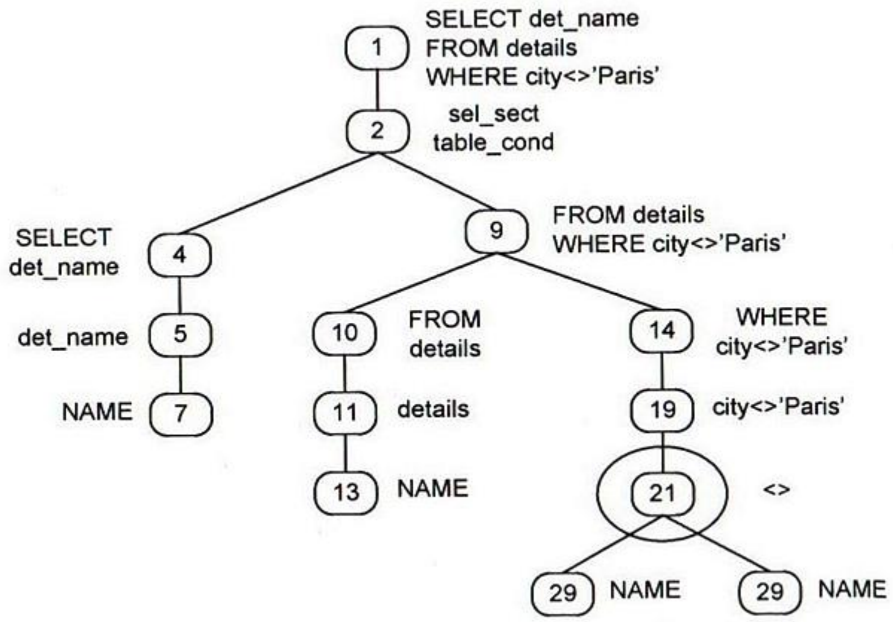

Fig. 3. The student's SQL-query parsing tree

To make a decision about a necessity of parsing trees traversal system needs information about similarity rate of the student's and reference queries. In this paper advanced q-gram method is proposed for queries similarity rate estimation.

Classical q-gram method. In this method, the original strings are broken into sets of overlapping word forms (q-gram) consisting of q symbols.

For example, in the case $q = 2$:

original string: abcdef (Len=6)
word forms: ab, bc, cd, de, ef (N=5)

in the case $q = 3$:

word forms: abc, bcd, cde, def (N=4)

Thus the number of q-grams for N strings of length Len, and a given q is described by following formula:

$$N = Len - q + 1. \qquad (1)$$

Each word form from the first set is compared with each word form from the second set. If the pair coincide, a comparison with the given word form from the second set is no longer produced. The similarity rate of two strings S might be considered as the ratio of C matches to the total number of word forms N:

$$S = C / N. \qquad (2)$$

Example 1. Assume that there are two strings representing SQL-queries (Len1 = Len2 = 19):

1) S1 = "SELECT color FROM p";
2) S2 = "SELECT owner FROM p".

For q=2 these queries will be represented as the following q-grams:

1) QG1={"SE" "EL" "LE" "EC" "CT" "T␣" "␣c" "co" "ol" "lo" "or" "r␣" "␣F" "FR" "RO" "OM" "M␣" "␣p" };

2) QG2={"SE" "EL" "LE" "EC" "CT" "T␣" "␣o" "ow" "wn" "ne" "er" "r␣" "␣F" "FR" "RO" "OM" "M␣" "␣p" }.

The count of word forms: $N = N1 = N2 = 18$. Obtained sets of word forms match in 13 positions. The similarity rate according to (2): $S = C\_(QG1, QG2) / N = 13/18 = 0.72$.

For the case when strings are of different lengths there is a question, which one of the two numbers of the word forms should be stated in the formula (2). It is obvious, if we compare the number of matches with the smaller of two values the similarity rate could be higher. In some cases it might be equal to 1 for the different strings. However, in the case of comparison with the greater value similarity rate could be too low for strings with minor differences. Thus the problem of choosing the reference value for comparison is quite principle.

Example 2. Let consider two strings with different lengths, which are SQL-queries:

1) S1="SELECT color, city FROM p", Len1 =25;
2) S2="SELECT color FROM p", Len2 =19.

For q = 2 these requests will be represented by the following q-grams:

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

1) QG1={“SE” “EL” “LE” “EC” “CT” “T␣” “␣c” “co” “ol” “lo” “or” “r,” “,␣” “␣c” “ci” “it” “ty” “y␣” “␣F” “FR” “RO” “OM” “M␣” “␣p” }, N1=24;

2) QG2={“SE” “EL” “LE” “EC” “CT” “T␣” “␣c” “co” “ol” “lo” “or” “r␣” “␣F” “FR” “RO” “OM” “M␣” “␣p” }, N2=18.

Given sets of word forms match in 17 positions: C(QG 1, QG 2) = 17 . In this case the similarity rate calculated by (2) with the grater word forms number is: $S = C\_(QG1, QG2)/N\_max = 17/24 = 0.708$. The similarity rate based on the smaller word forms number: $S = C\_(QG1, QG2)/N\_min = 17/18 = 0.94$.

Method based upon SQL keywords replacement. Any syntactically correct SQL-query contains a limited set of the certain keywords. Since many of the keywords might appear as in a correct query, and in an incorrect query to reduce their impact on the similarity assessment it makes sense to replace such keywords with the corresponding ID.

Example 3. Let consider a set of keywords: KW={“SELECT” “FROM” “WHERE”}.

There are 2 queries (Len1 = Len2 = 30):

1) S1="SELECT color FROM p WHERE a>10";

2) S2="SELECT color FROM t WHERE b>10";

1) QG1={“SE” “EL” “LE” “EC” “CT” “T␣” “␣c” “co” “ol” “lo” “or” “r␣” “␣F” “FR” “RO” “OM” “M␣” “␣p” “p␣” “␣W” “WH” “HE” “ER” “RE” “E␣” “␣a” “a>” “>1” “10”};

2) QG2={“SE” “EL” “LE” “EC” “CT” “T␣” “␣c” “co” “ol” “lo” “or” “r␣” “␣F” “FR” “RO” “OM” “M␣” “␣t” “t␣” “␣W” “WH” “HE” “ER” “RE” “E␣” “␣b” “b>” “>1” “10”}.

N1 = N2 = 29, C(QG1, QG2) = 25, $S = 25/29 = 0.86$.

After the keywords replacing:

1) S01="^0 color ^1 p ^2 a>10";

2) S02="^0 color ^1 t ^2 b>10".

Len01 = Len02 = 21.

1) QG1={“^0” “0␣” “␣c” “co” “ol” “lo” “or” “r␣” “␣^” “^1” “1␣” “␣p” “p␣” “␣^” “^2” “2␣” “␣a” “a>” “>1” “10”};

2) QG2={“^0” “0␣” “␣c” “co” “ol” “lo” “or” “r␣” “␣^” “^1” “1␣” “␣t” “t␣” “␣^” “^2” “2␣” “␣b” “b>” “>1” “10”}.

$C(QG1, QG\ 2) = 16$, $S = 16/21 = 0.76$.

Method based upon the parsing trees strings comparison. This method includes, firstly, strings construction for each syntactic tree and then their comparison.

Example 4. Let consider a parsing tree of the SQL-query «SELECT * FROM p WHERE City<>'Paris» (Table 1).

In tab.1: Node content – description of syntactic rules; Left index, Right index – indices of array elements for left and right descendants.

In order to distinguish a rule number, a child index and an ordinary numeric value, numbers could be marked with the following prefixes:

% – rule number;

– descendant index;

numeric values are stated without any additional marks.

The string obtained for the query:

S = “%1 #23 %2 #22 %7 #21 %9 #20 #9 %71 #19 #18 !E! %73 #17 %77 #16 %83 #15 #12 %99 #14 %25 #13 Paris %99 #11 %25 #10 City %10 #8 #2 %42 #7 %44 #6 %45 #5 %47 #4 %49 #3 p %11 #1 #0 list_of_sel_attrs: * !E! ”.

Further, constructed strings are compared by means of the classical q-gram method.

Presented methods for different types of similarity detection tasks give different output values. A promising challenge is to create a neural network and train it on the basis of student query series by giving as the input of the network the similarity values obtained by the three methods, and setting a required similarity rate as the output.

Conclusions

The approach proposed in the article allows providing comparison of the student's answer and reference query more efficiently then it is carried out in existing ITS for SQL.

A new approach to query's presentation in canonical form, as well as a scheme of the self-diagnosing method for the quality improvement of computer training in SQL were obtained.

The further investigations promising way is development of new task adaptive choice algorithms, depending on the current student progress, as well as interactive hints system creation.

Table 1

Parsing tree for SQL-query in example 4

№	***Element number***	***Rule number (%)***	***Left index (#)***	***Right index (#)***
0	!E!	12	-1	-1
1	"list_of_sel_attrs: *"	14	-1	-1
2	"sel_section: SELECT <opt_distinct> <list_of_sel_attrs>"	11	1	0
3	"p"	24	-1	-1
4	"table_decl: <tab_col_pse>"	49	3	-1
5	"from_attr: <table_decl>"	47	4	-1
6	"list_of_from_attrs: <from_attr>"	45	5	-1
7	"from_attrs: <list_of_from_attrs>"	44	6	-1
8	"from_section: FROM <from_attrs>"	42	7	-1
9	"un_sel_from_sect: <sel_section> <from_section>"	10	8	2
10	"City"	24	-1	-1
11	"column: <tab_col_pse>"	25	10	-1
12	"expression: <column>"	99	11	-1
13	"Paris"	24	-1	-1
14	"column: <tab_col_pse>"	25	13	-1
15	"expression: <column>"	99	14	-1
16	"statement: <expression> <> <expression>"	83	15	12
17	"logical_expression: <statement>"	77	16	-1
18	"where_section: WHERE <logical_expression>"	73	17	-1
19	"!E!"	103	-1	-1
20	"un_where_by_sect: <where_section> <by_section>"	71	19	18
21	"unary_select: <un_sel_from_sect> <un_where_by_sect>"	9	20	9
22	"sel_query: <unary_select>"	7	21	-1
23	"sql: <sql_query>"	2	22	-1
24	"root: <sql>"	1	23	-1

References (GOST 7.1:2006)

1. ElectronixTutor: an intelligent tutoring system with multiple learning resources for electronics [Electronic resource] / A. C. Graesser, X. Hu, B. D. Nye et al. // International Journal of STEM Education. – 2018. – Vol. 5. – Article No.: 15. DOI: 10.1186/s40594-018-0110-y.

2. The effectiveness of intelligent tutoring systems on K-12 students' reading comprehension: a meta-analysis [Text] / Z. Xu, K. Wijekumar, G. Ramirez, X. Hu, R.Irey // British Journal of Educational Technol-

ogies. – 2019. – Vol. 50, No. 6. – P. 3119–3137. DOI: 10.1111/bjet.12758.

3. Интеллектуальная компьютерная поддержка обучения составлению алгоритмов и SQL-запросов [Текст] : монография / А. С. Кулик, А. Г. Чухрай, Е. В. Гавриленко, Д. А. Гайдачук, И. А. Кулик. – Х. : Нац. аерокосм. ун-т ім. М. Є. Жуковського «Харків. авіац. ін-т», 2020. – 192 с.

4. Доценко, С. І. Інтелектуальні системи: принцип евристичної самоорганізації [Текст] / С. І. Доценко // Радіоелектронні і комп’ютерні системи. – 2020. – № 1(93). – С. 4-16. DOI: 10.32620/reks.2020.1.01.

5. Доценко, С. І. Інтелектуальні системи: принципи евристичної самоорганізації процесів смислового мислення та смислової діяльності [Текст] / С. І. Доценко // Радіоелектронні і комп’ютерні системи. – 2020. – № 2(94). – С. 4-22. DOI: 10.32620/reks.2020.2.01.

6. Mitrovic, A. A comparative analysis of cognitive tutoring and constraint-based modelling [Text] / A. Mitrovic, K. R. Koedinger, B. Martin // Proceedings of the Ninth International Conference on User Modeling. – Berlin: Springer-Verlag, 2003. – Vol. 2702 – P. 313-322.

7. Mitrovic, A. An intelligent SQL tutor on the web [Text] / A. Mitrovic // International Journal of Artificial Intelligence in Education. – 2003. – Vol. 13 – P. 173-197. HAL Id: hal-00197316.

8. Brass, S. Semantic Errors in SQL Queries: A Quite Complete List [Text] / S. Brass, C. Goldberg // Fourth International Conference on Quality Software (QSIC'04). – IEEE Computer Society, 2004. – P. 250-257.

9. VanLehn, K. The Behavior of Tutoring Systems [Text] / K. VanLehn // International Journal of Artificial Intelligence in Education. – 2006. – Vol. 16, No. 3. – P. 227-265.

10. Mayo, M. CAPIT: An Intelligent Tutoring System for Capitalisation and Punctuation [Text] / M. Mayo, A. Mitrovic, J. McKenzie // In Advanced Learning Technology: Design and Development Issues / C. Kinshuk, T. Okamoto (Eds.). – 2000. – P. 151-154. DOI: 10.1109/IWALT.2000.890594.

11. Mitrović, A. NORMIT: A Web-enabled tutor for database normalization [Text] / A. Mitrovic // Proceedings of the Int. conf. on computers in education, 2002. – Los Alamitos, CA. – IEEE Computer Societ. – P. 1276-1280. DOI: 10.1109/CIE.2002.1186210.

12. Giannandrea, L. A literature review on Intelligent Tutoring Systems and on student profiling [Text] / L. Giannandrea, M. Sansoni // Learning & Teaching with Media & Technology. – 2013. – P. 287-294.

13. Brass, S. Semantic errors in SQL queries: A quite complete list [Text] / S. Brass, C. Goldberg // Journal of Systems and Software. – 2006. – Vol. 79, No. 5. – P. 630-644. DOI: 10.1016/j.jss.2005.06.028.

14. Prior, J. The backwash effect on SQL skills grading [Text] / J. Prior, R. Lister // Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education, 2004. – Leeds, UK. – P. 32-36. DOI: 10.1145/1007996.1008008.

15. SQLator – an online SQL learning workbench [Text] / Sh. Sadiq, M. Orlowska, W. Sadiq, J. Lin // Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education, 2004. – Leeds, UK. – P. 223-227. DOI: 10.1145/1026487.1008055.

16. Dekeyser, S. Computer Assisted Assessment of SQL Query Skills [Text] / S. Dekeyser, M. De Raadt, T. Lee // Proceedings of the Eighteenth Australasian Database Conference, ADC 2007. – Ballarat, Victoria, Australia. – P. 53-62.

17. Hassan, N. Learning SQL through Intelligent Tutoring System [Text] / N. Hassan, M. Malik, M. Abd Wahab // Proceedings of E-Community Research Center National Conference on Bridging Digital Divide: Malaysia Initiatives, 2007. – Kuala Lumpur, Malaysia. – P. 1-10.

References (BSI)

1. Graesser, A. C., Hu, X., Nye, B. D. et al. ElectronixTutor: an intelligent tutoring system with multiple learning resources for electronics. *International Journal of STEM Education*, 2018, vol. 5. Article No.: 15. DOI: 10.1186/s40594-018-0110-y.

2. Xu, Z., Wijekumar, K., Ramirez, G., Hu, X., Irey, R. The effectiveness of intelligent tutoring systems on K-12 students’ reading comprehension: a meta-analysis. *British Journal of Educational Technologies*, 2019, vol. 50, no. 6, pp. 3119-3137. DOI: 10.1111/bjet.12758.

3. Kulik, A. S., Chukhray, A. G., Gavrilenko, Ye. V., Gaydachuk, D. A., Kulik, I. A. *Intellektual'naya komp'yuternaya podderzhka obucheniya sostavleniyu algoritmov i SQL-zaprosov* [Intelligence computer support for training in the algorithms and SQL queries development]. Kharkiv, National Aerospace University “Kharkiv Aviation Institute” Publ., 2020. 192 p.

4. Dotsenko, S. I. Intelektual'ni systemy: pryntsyp evrystychnoyi samoorhanizatsiyi [Intelligent systems: principle of heuristic self-organization]. *Radioelektronni i komp'uterni sistemi – Radioelectronic and computer systems*, 2020, no. 1(93), pp. 4-16. DOI: 10.32620/reks.2020.1.01.

5. Dotsenko, S. I. Intelektual'ni systemy: pryntsypy evrystychnoyi samoorhanizatsiyi protsesiv smyslovoho myslennya ta smyslovoyi diyal'nosti [Intellectual systems: principles of the heuristic self-organization of the processes of sense thinking and sense activity]. *Radioelektronni i komp'uterni sistemi – Radioelectronic and computer systems*, 2020, no. 2(94), pp. 4-22. DOI: 10.32620/reks.2020.2.01.

6. Mitrovic, A., Koedinger, K. R., Martin, B. A comparative analysis of cognitive tutoring and constraint-based modellin. *Proceedings of the Ninth International Conference on User Modeling*, Berlin, "Springer-Verlag" Publ., 2003, vol. 2702, pp. 313-322.

7. Mitrovic, A. An intelligent SQL tutor on the web. *International Journal of Artificial Intelligence in Education*, 2003, vol. 13, pp. 173-197. HAL Id: hal-00197316.

8. Brass, S., Goldberg, C. Semantic Errors in SQL Queries: A Quite Complete List. *Fourth International Conference on Quality Software (QSIC'04)*, IEEE Computer Society, 2004, pp. 250-257.

9. VanLehn, K. The Behavior of Tutoring Systems. *International Journal of Artificial Intelligence in Education*, 2006, vol. 16, no. 3, pp. 227-265.

10. Mayo, M., Mitrovic, A., McKenzie, J. CAPIT: An Intelligent Tutoring System for Capitalisation and Punctuation. In *Advanced Learning Technology: Design and Development Issues*, 2000, pp. 151-154. DOI: 10.1109/IWALT.2000.890594.

11. Mitrović, A. NORMIT: A Web-enabled tutor for database normalization. *Proceedings of the Int. conf. on computers in education ICCE 2002*, Los Alamitos, CA. IEEE Computer Society, pp. 1276-1280. DOI: 10.1109/CIE.2002.1186210.

12. Giannandrea, L., Sansoni, M. A literature review on Intelligent Tutoring Systems and on student profiling. *Learning & Teaching with Media & Technology*, 2013, pp. 287-294.

13. Brass, S., Goldberg, C. Semantic errors in SQL queries: A quite complete list. *Journal of Systems and Software*, 2006, vol. 79, no. 5, pp. 630-644. DOI: 10.1016/j.jss.2005.06.028.

14. Prior, J., Lister, R. The backwash effect on SQL skills grading. *Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education*, Leeds, UK, June 2004, pp. 32-36. DOI: 10.1145/1007996.1008008.

15. Sadiq, Sh., Orlowska, M., Sadiq, W., Lin, J. SQLator – an online SQL learning workbench. *Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education*, Leeds, UK, June 2004, pp. 223-227. DOI: 10.1145/1026487.1008055.

16. Dekeyser, S., De Raadt, M., Lee, T. Computer Assisted Assessment of SQL Query Skills. *Proceedings of the Eighteenth Australasian Database Conference, ADC 2007*, Ballarat, Victoria, Australia, January - February, 2007, pp. 53-62.

17. Hassan, N., Malik, M., Abd Wahab, M. Learning SQL through Intelligent Tutoring System. *Proceedings of E-Community Research Center National Conference on Bridging Digital Divide: Malaysia Initiatives*, Kuala Lumpur, Malaysia, 2007, pp. 1-10.

Надійшла до редакції 15.04.2021, розглянута на редколегії 20.05.2021

МЕТОД АНАЛІЗУ ЗАПИТІВ СТУДЕНТА ПРИ ІНТЕЛЕКТУАЛЬНОМУ КОМП'ЮТЕРНОМУ НАВЧАННІ SQL

А. Г. Чухрай, О. В. Гавриленко

Останнім часом одними з найбільш затребуваних на світовому ринку праці спеціальностей стали IT-спеціальності. Разом з тим традиційне навчання в умовах масового виробництва навіть при наявності дуже хорошого педагога має істотний недолік - принципову неможливість адаптуватися до кожного, кого навчають. З 60-х рр. XX століття дослідники в усьому світі розробляють різні комп'ютерні засоби навчання, що володіють, в тій чи іншій мірі, адаптивними функціями. Проте, завдання розробки досконалого комп'ютерного педагога ще далеке від свого вирішення. **Предметом** дослідження в статті є процес аналізу запитів студента під час інтелектуального комп'ютерного навчання мові SQL. **Метою** є розробка методу аналізу SQL запитів студента. **Завдання**: на основі принципів технічної діагностики і методів лексичного, синтаксичного аналізу комп'ютерних програм сформувати загальну схему і особливості методу аналізу SQL запитів студента. Розробити методи побудови дерева синтаксичного розбору. Створити методи зіставлення еталонного і реального SQL запитів за ступенем їх схожості. Проілюструвати функціонування розроблених методів на конкретних прикладах. Використовуваними **методами** є: методи автоматичного тестування комп'ютерних програм, методи лексичного і синтаксичного аналізу комп'ютерних програм, методи побудови дерев синтаксичного розбору комп'ютерних програм, методи діагностування об'єктів на основі порівняння з еталоном, методи аналізу рядків, метод q-грам. Одержані наступні **результати**: сформований метод аналізу SQL запи-

 ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

тів студента під час інтелектуальних комп'ютерного навчання умінням складати SQL запити, що ґрунтується на системному підході, що включає автоматичне тестування на реальних даних, побудову дерев синтаксичного розбору запитів, зіставлення з еталоном і комплексне визначення ступеня схожості запитів. **Висновки**. Наукова новизна полягає в отриманні подальшого розвитку методу аналізу SQL запитів студента під час інтелектуального комп'ютерного навчання умінням складати SQL запити.

Ключові слова: інтелектуальна комп'ютерна система; структурована мова запитів; навчання; аналіз запитів; дерево синтаксичного розбору.

МЕТОД АНАЛИЗА ЗАПРОСОВ СТУДЕНТА ПРИ ИНТЕЛЛЕКТУАЛЬНОМ КОМПЬЮТЕРНОМ ОБУЧЕНИИ SQL

А. Г. Чухрай, Е. В. Гавриленко

В последнее время одними из наиболее востребованных на мировом рынке труда специальностей стали IT–специальности. Вместе с тем традиционное обучение в условиях массового производства даже при наличии очень хорошего педагога обладает существенным недостатком – принципиальной невозможностью адаптироваться к каждому обучаемому. С 60-х гг. XX столетия исследователи во всем мире разрабатывают различные компьютерные средства обучения, обладающие, в той или иной степени, адаптивными функциями. Тем не менее, задача разработки совершенного компьютерного педагога еще далека от своего разрешения. **Предметом** исследования в статье является процесс анализа запросов обучаемого во время интеллектуального компьютерного обучения языку SQL. **Целью** является разработка метода анализа SQL запросов обучаемого. **Задачи**: на основе принципов технической диагностики и методов лексического, синтаксического анализа компьютерных программ сформировать общую схему и особенности метода анализа SQL запросов обучаемого. Разработать методы построения дерева синтаксического разбора. Создать методы сопоставления эталонного и реального SQL запросов по степени их похожести. Проиллюстрировать функционирование разработанных методов на конкретных примерах. Используемыми **методами** являются: методы автоматического тестирования компьютерных программ, методы лексического и синтаксического анализа компьютерных программ, методы построения деревьев синтаксического разбора компьютерных программ, методы диагностирования объектов на основе сравнения с эталоном, методы анализа строк, метод q-грамм. Получены следующие **результаты**: сформирован метод анализа SQL запросов обучаемого во время интеллектуального компьютерного обучения умениям составлять SQL запросы, основывающийся на системном подходе, включающем автоматическое тестирование на реальных данных, построение деревьев синтаксического разбора запросов, сопоставление с эталоном и комплексное определение степени похожести запросов. **Выводы**. Научная новизна заключается в получении дальнейшего развития метода анализа SQL запросов обучаемого во время интеллектуального компьютерного обучения умениям составлять SQL запросы.

Ключевые слова: интеллектуальная компьютерная система; структурированный язык запросов; обучение; анализ запросов; дерево синтаксического разбора.

Чухрай Андрій Григорович – д-р техн. наук, доцент, зав. каф. математичного моделювання та штучного інтелекту, Національний аерокосмічний університет ім. М. Є. Жуковського «Харківський авіаційний інститут», Харків, Україна.

Гавриленко Олена Володимирівна – канд. техн. наук, доцент каф. систем управління літальних апаратів, Національний аерокосмічний університет ім. М. Є. Жуковського «Харківський авіаційний інститут», Харків, Україна.

Andrey Chukhray – Doctor of Technical Sciences, Associate Professor, Head of the Department mathematical modelling and artificial intelligence, National Aerospace University «Kharkiv Aviation Institute», Kharkiv, Ukraine,
e-mail: achukhray@gmail.com, ORCID: 0000-0002-8075-3664.

Olena Havrylenko – Candidate of Technical Science, Associate Professor of the Department «Fly Vehicle's Control Systems», National Aerospace University «Kharkiv Aviation Institute», Kharkiv, Ukraine,
e-mail: lm77191220@gmail.com, ORCID: 0000-0001-5227-9742.