Building Individual Educational Routes for Learning SQL Queries

Anton Govorov, Marina Govorova, Helen Slizen, Sergey Ivanov
ITMO University
Saint Petersburg, Russia
antongovorov@gmail.com

Abstract–**Due to how prevalent the usage of database technologies is in everyday life and related professional fields, the insufficient qualifications of specialists in relevant positions becomes an issue. The aim of this paper is to show the importance of forming individual educational routes as a factor influencing the effectiveness of training and the ability of a specialist to develop skills in professional activities. In learning SQL queries, repeatedly performing typical tasks, seeing the results, and adjusting queries accordingly, is necessary. Therefore, computer training simulators can be considered the most effective tool for learning SQL queries. This paper proposes a method for building an individual educational route for fulfilling tasks meant to train professional skills in compiling SQL queries. The task and learner models are described within the method. The A* algorithm is modified based on the proposed models, and the conditions of using it for creating an optimal individual educational route are described. Based on the theoretical study results, a computer simulator for training professional skills in writing SQL queries is implemented using the method of building individual educational routes for fulfilling tasks meant to train professional skills in writing SQL queries. Based on the analysis of the results of a test run, the conclusion is that the usage of the proposed method and the algorithm implemented in a computer simulator can be considered effective.**

I. Introduction

Currently, the relevance of using database technologies is explained by a constant increase in the volume of processed and stored information, as well as the modernization of information processes taking place in all areas of modern life. Databases are used by various systems to store the bulk of the accumulated information, and their maintenance requires highly qualified specialists [1], [2].

Computer training simulators can be considered the most effective tool that allows a learner to build professional skills in compiling SQL queries. This is due to the fact that when learning the SQL language, the learner needs to repeatedly perform typical tasks, see the result of his work, and adjust his queries if necessary.

Currently, some principles of computer simulators are used by various educational platforms that offer online courses on databases (for example, OpenEdu [3], edX [4], Stanford Online [5], National Open University INTUIT [6]). At the same time, they do not always provide the environment for forming sustainable practical skills in performing typical tasks in the professional sphere. A significant drawback of such platforms is that teachers are unable to change the course curriculum or its content, which makes it difficult to fully individualize the learning process [7].

The individualization of the learning process is a relevant field of study in the development of educational technologies. Applying an individual approach to training provides the learner with the opportunity to form his own educational route and increases the effectiveness of training. The learner can see his prospects at the stage of forming the training route, which can positively affect the learning outcome.

Marsova S.E. in the study [8] claims that “an individual educational route contributes to a more effective organization of training”. Petruneva R.M. believes that an individual trajectory can serve as a means of “enriching the educational process” [9]. According to Sivolapova A.K. and Gilmulina T.P., individual trajectories can allow learners to “successfully carry out professional activities” [10].

To implement individual educational trajectories and routes, many authors propose using graph processing methods. For example, in research [11], building an educational trajectory of a student in a higher educational institution using graph theory is considered. In [12], the authors use a directed graph to individualize the learning path as well. I.A. Alshalabi et al. [13] consider the problem of automated adaptive learning. Furthermore, in the study [14], the authors state that the problem of learning paths can be represented as a weighted graph.

Despite the large number of studies devoted to the problem of forming individual educational routes, a single approach to technology required to solve it does not exist. Solutions are based on the specifics of the subject area and the individual features of the problem facing the researcher.

In connection with the problems identified above, the authors of the paper have proposed presenting the training course of the learner in the form of a graph, where each vertex is a set of tasks. The aim was to construct an optimal educational route between the vertices of such a graph, taking into account the needs of a particular learner in the context of his professional growth.

For constructing an optimal educational route, the authors modified the A* algorithm and built a computer simulator for training professional skills in compiling SQL queries, motivating the learning process, and allowing the teacher to individualize it.

II. Construction of Mathematical Models of Learning Objects in an Information Learning System

The task of constructing an individual educational route for learners can be compared with the task of finding the shortest path between the vertices of a graph. The shortest path problem is the task of finding the shortest path between two vertices on a graph in which the sum of the weights of the edges making up the path is minimized.

The training course consists of topics, each of which contains sets of 3 tasks. In sets of one topic, tasks of various complexity of levels from 1 to 5 are stored. The overall complexity of the tasks in the set is designated as CP (Task complexity). A set is formed from tasks, and its complexity is set to equal to the sum of the complexities of all tasks in it. In one topic, sets with different complexities are possible.

Depending on the complexity of the tasks completed, the learner receives an assessment for the entire course. The teacher indicates MK (Minimum knowledge level), the focal assessment that the learner is applying for.

The task of the algorithm is to build a route between tasks taking into account the total complexity of tasks in it to achieve the focal level of skills. The algorithm adapts the individual educational route if the learner is unable to cope with the tasks assigned to him.

In the context of the task, the course model can be described as a weighted oriented graph *G (V, E)*, in which:

– *V={v_i}* – set of nodes, each of which corresponds to a set of three training tasks on a specific topic, $i = \overline{1,h}$, h – number of sets of assignments for all educational topics of the course;

– $E = \{(v_1, v_2), (v_2, v_3), \dots\}$ – set of edges comprised of node pairs (fig. 1).

The weights of the edges have the following meanings: 1 - the weight of the edge determined by the vertices corresponding to the task sets of one topic; 2 - the weight of the edge defined by the vertices corresponding to the sets of tasks of the topic and subtopic; 3 - the weight of the edge defined by the vertices corresponding to the task sets of this topic and the next topic.

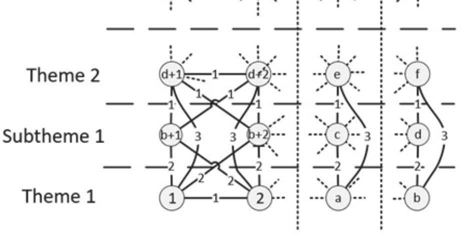

Fig. 1. Course Model

The weights of the edges between the vertices of one topic are 1. The weights of the edges from the vertices of the current topic to the vertices of the next topic are 3. The weights of the edges from the vertices of those to the vertices of their subtopics are equal 2. Sets have the following difficulty levels:

1) 3-5 points in total for tasks in the set – tasks for assessment 3;

2) 6-10 points in total for tasks in the set – tasks for assessment 4;

3) 11-15 points in total for tasks in the set – tasks for rating 5.

Certain topics can be designed to contain subtopics. Subtopics include a set of tasks that a learner is required to complete if he has not scored a certain points threshold when completing the main set of tasks for the topic.

If the learner has completed all the tasks from the set, he proceeds to the next topic. If the learner has completed two of the three tasks and is rated 3 or 4, he also proceeds to the tasks of the next topic. However, if he is rated 5, he will have to complete a set of tasks from a subtopic before starting a new topic.

Current academic performance in the course is expressed as a percentage of the tasks that the learner could complete if he completed all the tasks perfectly.

A. Building models of educational objects

The learner and the training task are considered educational objects. The analysis of the subject area allows us to distinguish the following parameters of the models of educational objects.

For the training task:

1. ct – task complexity, $ct_i \in \{1,2,3,4,5\}, i = \overline{1,n};$ n – number of tasks in the course. The difficulty is indicated by the teacher when creating the training task.

2. The degree of relationship of tasks to topics within the course framework can be represented in the form of a matrix $T_{n*m} = \{t_{ij}\}$; $i = \overline{1,n}$; $j = \overline{1,m}$ (formula 1):

$$T_{n \times m} = \begin{bmatrix} t_{11} & \cdots & t_{1m} \\ \vdots & \ddots & \vdots \\ t_{n1} & \cdots & t_{nm} \end{bmatrix} \qquad (1)$$

where:

T (Task list) – matrix of tasks to topics assignments with a dimension $n \times m$;

t_{ij} – degree of compliance of task *i* with topic *j*, $0 \leq t_{ij} \leq 1$; $\sum_{j=1}^{m} t_{ij} = 1$; *i*=$\overline{1,n}$; *j*=$\overline{1,m}$;

n – number of tasks in the course;

m – number of topics.

For the learner:

1. mk – average minimum knowledge level needed for the learner ($0{,}6 \leq mk \leq 1$).

2. *RIT = [rit$_1$, rit$_2$, ..., rit$_m$]* – array of results of the learner's testing reflecting the degree of mastery of topics ($0 \leq rit_i \leq 1$, $i=\overline{1,m}$).

3. *LT = [lt$_1$, lt$_2$, ..., lt$_m$]* – level of knowledge on academic topics ($lt_q \in (0;1]$, $q = \overline{1,m}$) (formula 2):

$$lt_q = \frac{\sum_{i=1}^{u}(a_i * c_i)}{5u} + \sum_{j=q+1}^{m} \frac{\sum_{i=1}^{e_j}(b_k * d_k)}{5e_j} \qquad (2)$$

where:

u – number of tasks in the set, $u = 3$;

q – topic number, $q = \overline{1,m}$;

a_i –degree of attitude of the i-th task to the topic from the current topic q, $a_i \in \{t_{pq}\}, p = \overline{1,n}$;

c_i – task complexity a_i, $c_i \in \{ct_p\}, p = \overline{1,n}$;

e_j –number of tasks in topic set j corresponding to the topic q, $e_j \in \{1, 2, 3\}$;

b_k –degree of relation of the k-th task in the set of topics j to theme q, $j = \overline{q+1,m}$; $k = \overline{1,e_j}$, $b_k \in \{t_{pq}\}, p = \overline{1,n}$;

d_k – task complexity b_k, $d_k \in \{ct_p\}, p = \overline{1,n}$;

$a_i \neq b_k; i = \overline{1,3}, k = \overline{1,3}$.

For the last topic of the course, the second term in formula (2) is reset.

4. *PT = [pt$_1$, pt$_2$, ..., pt$_m$]* – list of priority topics ($0 \leq pt_i \leq 1$, $i=\overline{1,m}$. Default: $pt_i = 1 - rt_i$.

The considered parameters of the learner and study task models are used in the algorithms for constructing educational routes discussed below.

B. Formation of task sets

In the context of performing tasks on programming or writing SQL queries, situations are possible where a practical, test or laboratory task refers not to one, but to several topics at once (formula 1).

As stated previously, the tasks are combined into sets of three tasks. qt_q –maximum number of sets on q is determined by the formula (3):

$$qt_q = \frac{g!}{3!\,(g-3)!} \qquad (3)$$

where:

q – topic number, $q = \overline{1,m}$;

g – number of tasks on the topic q, $3 \leq g \leq n$.

The method provides the ability for the teacher to change to delete sets of tasks after the formation of all possible combinations of tasks on the current topic. Each set has its complexity calculated by the formula (4):

$$cp_j^q = \sum_{i=1}^{3} c_i \qquad (4)$$

where:

q – topic number, $q = \overline{1,m}$;

cp_j^q – complexity of the j-th set from theme q; $j = \overline{1,qt_q}$;

c_i – complexity of the i-th task in the current theme set q, $i = \overline{1,3}$, $c_i \in \{ct_p\}$, $p = \overline{1,n}$.

The complexity of the set is used to build an educational route using the modified A* algorithm, which is discussed later. A distribution of tasks is possible in the model that allows to increase the degree of coverage of the topic when completing the next task on the current topic. Table I shows an example of the ratio of topics and tasks.

TABLE I. AN EXAMPLE OF THE RELATIONSHIP BETWEEN TOPICS AND TASKS

No.	Theme 1	Theme 2	Theme 3
Task 1		1	
Task 2	1		
Task 3		0.2	0.8

In this example, assignment No. 1 relates only to topic No. 2, assignment No. 2 relates only to topic No. 1, assignment No. 3 relates to both topic No. 2 and topic No. 3, and the degree of attitude to topic No. 2 is 0.2, and to the topic No. 3 - 0.8. It should be noted that, performing the rear 3 in the framework of work on topic No. 3, the learner trains skills not only on the current topic, but also on the previous topic.

When using an array reflecting the degree of coverage of topics, tasks related to the least mastered topics can be selected. The method assumes that the degree to which the tasks are related to the topic is determined by the teacher when creating the task.

It should be noted that task number 3 cannot be used when studying topic number 2, since topic number 3 is not yet learnt.

When developing the tasks on topics, the teacher first sets the weight of the tasks t_{ij} so that for each task their sum is equal to 1 (formula 5):

$$\sum_{j=1}^{k} t_{ij}^{p} = 1 \qquad (5)$$

where:

t_{ij}^{p}– weight coefficient of the degree of attitude of the task to the topic; i – course assignment number, $i = \overline{1,n}$; j – course topic number, $j = \overline{1,m}$; k – number of topics to which the task is related, $1 \leq k \leq m$; p – task number in set, $1 \leq p \leq 3$; $t_{ij} \in T$ (formula 1).

As each set has 3 tasks, the total weight of the set for all topics is 3.

III. The method of building individual educational training routes for learning professional skills in writing SQL queries

The A* algorithm is a modification of Dijkstra's algorithm, optimized for a single end vertex. Dijkstra's algorithm can find paths to all vertices, while A* finds a path to one vertex, prioritizing those that provide a shorter path to the goal.

In the modified A* algorithm, the cost of the path to tasks from the same topic has to be less than the cost of the path to tasks from the next topic. The cost of the path from the subtopic tasks will also be less than the cost of the path to the tasks from the next topic. The ability to search for paths in sets of tasks from the current topic and subtopic depends on the results and goals of the learner.

Taking into account the models of the training task and the learner, it is possible to highlight the functional features necessary to modify the algorithm for finding the route of task execution:

1) Tasks have a weight that reflects the degree of difficulty. Tasks are combined into sets. The weight of the set is equal to the sum of the weights of the tasks.

2) Themes consist of sets of tasks. One task may be related to different topics.

3) Topics can have subtopics that also consist of task sets.

4) At the beginning of the course, the learner or teacher sets the target grade. The target assessment displays the minimum level of difficulty of the tasks for the learner to complete.

In Dijkstra's algorithm, the front of the search for the optimal path expands in all directions. However, a search is usually performed for only one vertex.

The implementation of the A* algorithm differs from the implementation of the Dijkstra's algorithm only by the presence of a heuristic function, the purpose of which is to adapt the route to repeat the least mastered topics, or topics with higher priority.

In the operation of the A* algorithm, the function of the cost of the path to the top for the vertexes is calculated *v* (formula 6):

$$f(v)=g(v)+h(v) \qquad (6)$$

where:

g(v) – estimated cost of the path to the vertex v from the starting vertex;

h(v) – heuristic approximation of the cost of the path from the vertex v to the final goal.

Function *f(v)* is the length of the path to the goal, which is the sum of the distance traveled *g(v)* and a heuristic function that characterizes the distance from the current vertex to the final one *h(v)*. Based on this, the smaller the value of *f(v)*, the earlier the vertex v opens.

Algorithm A* acts like Dijkstra's algorithm by searching all the routes leading to the goal. First it considers those that based on the results of the heuristic function are seen as the most desirable.

Dividing the parameters of the student's models, course and tasks into those that need to be used in the main algorithm, and those that need to be taken into account only when implementing the heuristic function, is necessary for implementing the algorithm. The following parameters were selected as parameters of the heuristic function:

1) LT –degree of coverage of topics, determined for each topic by the formula 2. The model of each of the tasks implies the presence of a parameter that reflects the assignment of the task to certain educational topics.

2) PT – priority of the topic.

Depending on the significance of priorities, the algorithm calculates the value of the heuristic function according to the following formula (7):

$$f(x) = 1{,}98^{x^4} - 1 \qquad (7)$$

where:

$x \in [0;1]$; $x = pt_i - lt_i$; lt_i – topic coverage; pt_i – priority for learning topics, $i = \overline{1,m}$.

The function graph is shown in Fig. 2.

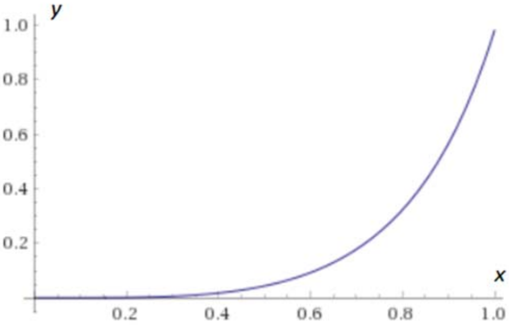

Fig. 2. Heuristic Function Graph

The graph shows that in the case of insufficient level of development of the topic and its high significance, taking into account its priority (value along the x axis), the value of the heuristic function for this topic will tend to unity (value along the y axis).

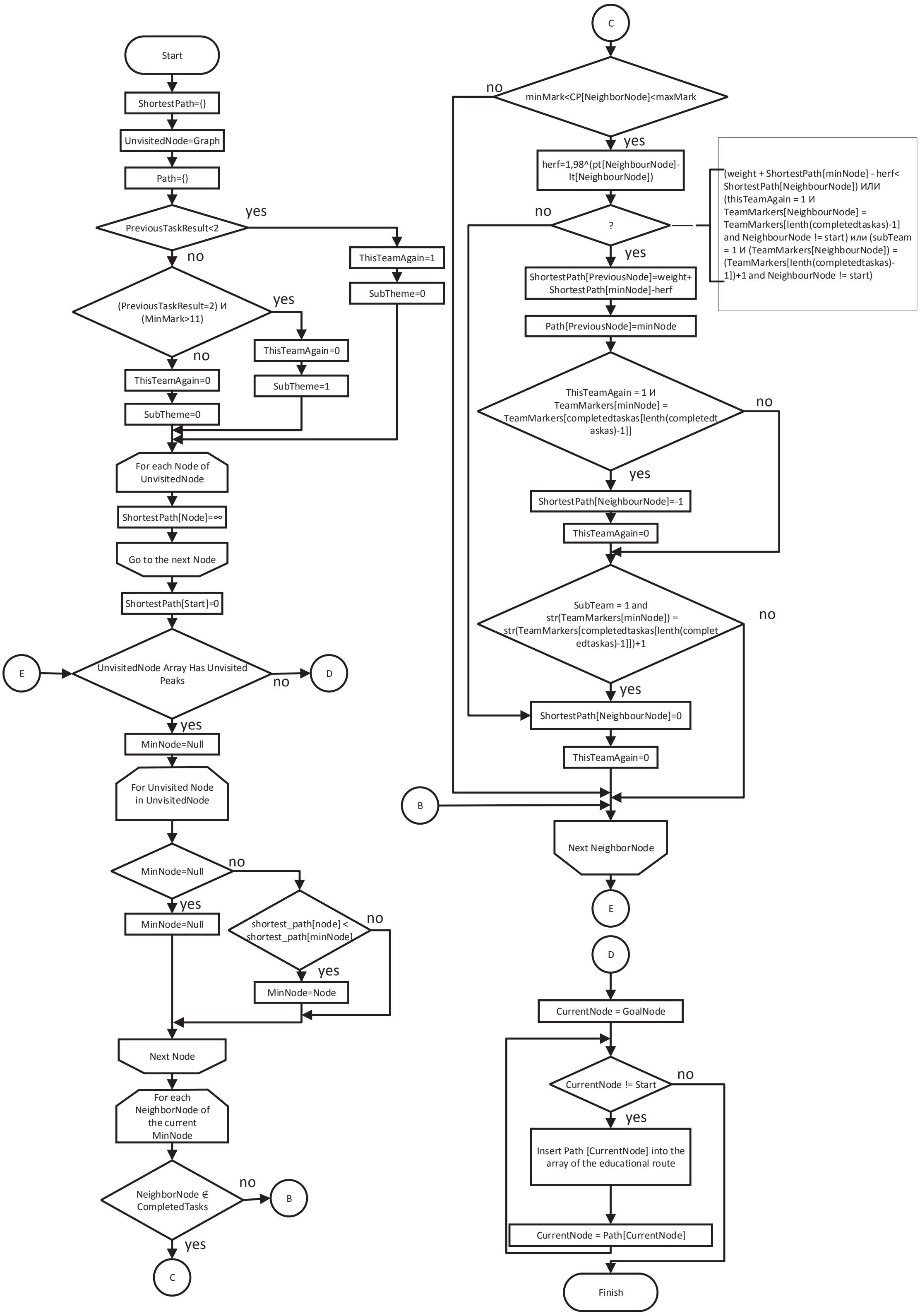
Start
ShortestPath={}
UnvisitedNode=Graph
Path={}
PreviousTaskResult<2
yes
ThisTeamAgain=1
SubTheme=0
no
(PreviousTaskResult=2) И (MinMark>11)
yes
ThisTeamAgain=0
SubTheme=1
no
ThisTeamAgain=0
SubTheme=0
For each Node of UnvisitedNode
ShortestPath[Node]=∞
Go to the next Node
ShortestPath[Start]=0
E
UnvisitedNode Array Has Unvisited Peaks
no
D
yes
MinNode=Null
For Unvisited Node in UnvisitedNode
MinNode=Null
no
yes
MinNode=Null
shortest_path[node] < shortest_path[minNode]
no
yes
MinNode=Node
Next Node
For each NeighborNode of the current MinNode
NeighborNode ∉ CompletedTasks
no
B
yes
C
C
no
minMark<CP[NeighborNode]<maxMark
yes
herf=1,98^(pt[NeighbourNode]-lt[NeighbourNode])
no
?
(weight + ShortestPath[minNode] - herf< ShortestPath[NeighbourNode]) ИЛИ (thisTeamAgain = 1 И TeamMarkers[NeighbourNode] = TeamMarkers[lenth(completedtaskas)-1] and NeighbourNode != start) или (subTeam = 1 И (TeamMarkers[NeighbourNode]) = (TeamMarkers[lenth(completedtaskas)-1])+1 and NeighbourNode != start)
yes
ShortestPath[PreviousNode]=weight+ ShortestPath[minNode]-herf
Path[PreviousNode]=minNode
ThisTeamAgain = 1 И TeamMarkers[minNode] = TeamMarkers[completedtaskas[lenth(completedtaskas)-1]]
no
yes
ShortestPath[NeighbourNode]=-1
ThisTeamAgain=0
SubTeam = 1 and str(TeamMarkers[minNode]) = str(TeamMarkers[completedtaskas[lenth(completedtaskas)-1]])+1
no
yes
ShortestPath[NeighbourNode]=0
ThisTeamAgain=0
B
Next NeighborNode
E
D
CurrentNode = GoalNode
CurrentNode != Start
no
yes
Insert Path [CurrentNode] into the array of the educational route
CurrentNode = Path[CurrentNode]
Finish

Fig. 3. Scheme of the modified A* algorithm

The modified A* algorithm uses the following parameters:

Node – graph vertex.

MinNode – the vertex from which path construction options are considered.

NeighborNode – the vertex of the graph adjacent to the current *MinNode*.

ShortestPath – array of minimum path costs from the initial vertex of the graph to any vertex of the graph.

Weight – the cost of moving from one vertex of the graph to the next.

AllPath – an array of connections between vertexes storing all existing optimal paths (having a minimum transition cost) to any vertex of the graph.

PreviousTaskResult – the result of the previous set of tasks.

UnvisitedNode – source graph in which each vertex corresponds to a set of tasks on a specific topic.

Start – vertex corresponding to the last completed set of tasks.

CompletedTasks – array of vertexes corresponding to completed task sets.

MinMark – the minimum level of difficulty of the tasks corresponding to the target level of mastering the course, which corresponds to *mk.*

MaxMark – the maximum level of difficulty of tasks corresponding to the target level of mastering the course.

GoalNode – any of the tops of the last topic corresponding to the level of the target rating.

CP – array of complexity of all task sets in the graph.

ThisTeamAgain – marker of the need for repeated passage of tasks of the topic.

SubTheme – marker for completing subtopics.

LT – the degree of coverage of topics, determined for each topic.

PT – priority of the topic.

herf – heuristic function.

Building a route for completing task sets based on a modified A* algorithm allows to:

– re-complete the set of tasks for the same topic if the set of tasks on the topic is unsatisfactory (less than two tasks),

– with satisfactory completion of the current set (2 tasks) and an average or high target rating, complete a set of tasks from a subtopic of the same difficulty level or perform a different set of tasks of the same difficulty level from the current topic to maintain the target assessment level,

– ensure compliance of the complexity of the current task set with the target assessment,

– change the cost of the path, taking into account the application of the heuristic function described above.

IV. Results and discussion

Given the list of functional features of means of increasing students' motivation, described in the paper [15], the following features for implementing the system were identified:

– interfaces for providing video content (lectures, mini-lectures) in learning management systems,

– an interface for comments on practical tasks,

– a reward system in the learning process with badges, achievements and awards,

–a system of levels that reflects the progress of the student in the learning process.

The course “Using the SELECT statement” has been developed. For the course, three training databases and 187 tasks for them were developed for the following topics:

1) Automated control system for design and technological design.

2) Educational activities of the university.

3) Computer RPG indie game (role-playing game, a computer game based on the principles of board games).

Experimental testing of the research results in the process of training students of the Department of Intelligent Technologies in Humanities (ITH) (Fig. 4, 5) and students of the Faculty of Secondary Professional Education (SPE) (Fig. 6, 7) of ITMO University was carried out using the developed software package.

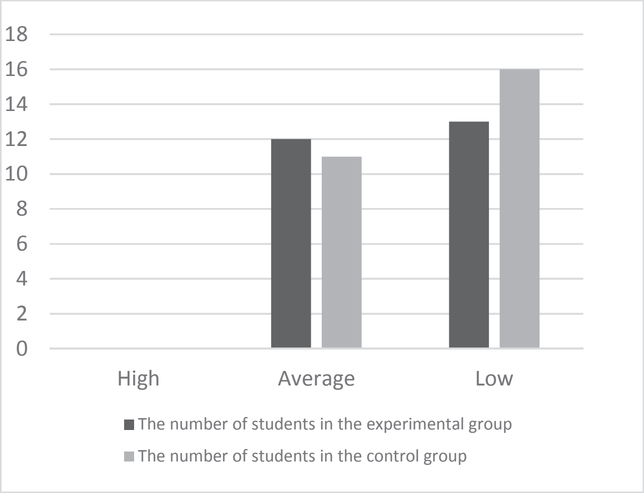

Fig. 4. The results of measuring the level of skills of students of the ITH Department before the experiment

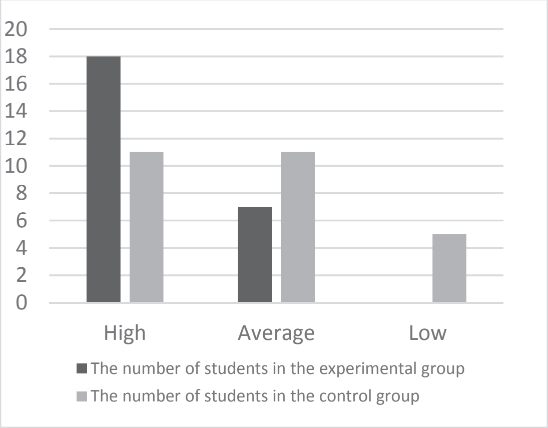

Fig. 5. The results of measuring the level of skills of students of the ITH Department after the experiment

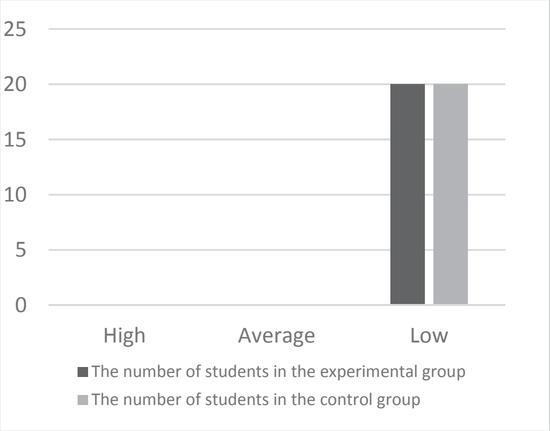

Fig. 6. The results of measuring the level of skills of students of the SPE faculty before the experiment

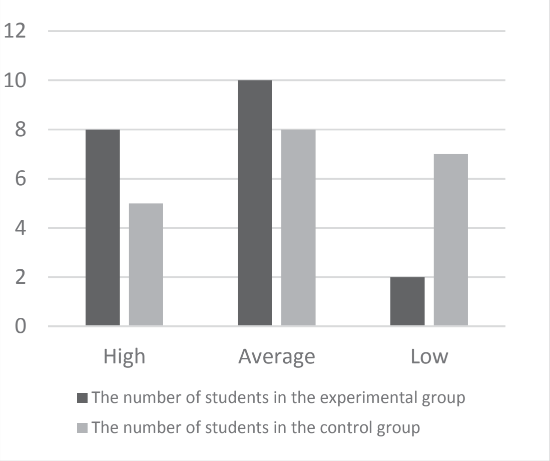

Fig. 7. The results of measuring the level of skills of students of the SPE faculty after the experiment

The test results confirmed the main hypothesis of the experiment on improving the effectiveness of training students in IT, and allowed to conclude the effectiveness of using the proposed method and the algorithm implemented in a computer simulator.

V. Conclusion

At this stage of the research, a method has been developed to build individual educational routes for practical tasks in order to train professional skills in compiling SQL queries.

The results of the theoretical study made it possible to design and implement a computer simulator for training professional skills in composing SQL queries using gamification principles.

The developed computer simulator can be used by higher and secondary professional educational institutions, as well as corporate training systems for retraining and advanced training, in the educational processes of training specialists in areas related to the development and maintenance of automated systems that require qualifications in the field of databases.

References

[1] Shvetsov D. "Intelligent data storage systems in process control systems", *Modern automation technologies*, vol. 4, 2011, pp. 42-46.

[2] Burakov P.V., Petrov V.Yu, *Introduction to database systems.* St. Petersburg: Textbook, 2010.

[3] The course "Data Management" from FSAEI HE "St. Petersburg Polytechnic University of Peter the Great", *National platform for open education*, Web: https://openedu.ru/course/spbstu/DATAM/

[4] Developing SQL Databases, *EDX official website*, Web: https://www.edx.org/course/developing-sql-databases-microsoft-dat215-1x-2

[5] Databases: Self Pased, *Stanford University web-site*, Web: https://lagunita.stanford.edu/courses/DB/2014/SelfPaced/about

[6] Course "Databases", *National Open University "INTUIT"*, Web: https://www.intuit.ru/studies/courses/508/364/info

[7] Zolotukhin S.A. "Advantages and disadvantages of massive open online courses", *Discussion*, vol. 4 (56), 2015, pp. 97-102.

[8] Marsova S.E. "The specifics of the implementation of the individual educational route at the master's level", *Political Mathematical Network Electronic Scientific Journal of the Kuban State Agrarian University*, vol. 85 (01), 2013, pp. 1-12.

[9] Petruneva R.M. "The individually-oriented organization of the educational process: illusions and reality", *Higher Education in Russia*, vol. 5, 2011, pp. 65-70.

[10] Sivolapova A.K., Gilmulina T.P. "Tools for building an individual educational trajectory of a student", *Vocational education in Russia and abroad*, vol. 3 (23), 2016, pp. 79-81.

[11] Maltekbasov M.Zh., Prokofieva M.A., Eskendirov B.N., Nurbosynova G.S. "Features of the application of graph theory in the design of educational trajectories at a university", *International Journal of Experimental Education*, vol. 1-1, 2015, pp. 102-105.

[12] Guillaume Durand, Nabil Belacel, François LaPlante. "Graph theory based model for learning path recommendation", *Information Sciences*, vol. 251, 2013. pp. 10-21.

[13] Ibrahim Alkore Alshalabi, Samir Hamada, and Khaled Elleithy, “Automated Adaptive Learning using Smart Shortest Path Algorithm for Course Units,” *IEEE Long Island Systems, Applications and Technology LISAT 2015*, Long Island, NY, May 2015.

[14] Muhammad, A., Zhou, Q., Beydoun, G., Xu, D. & Shen, J. "Learning path adaptation in online learning systems", *20th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 2016, pp. 421-426. United States: IEEE.

[15] Govorov A.I., Govorova M.M., Valitova Yu.O. "Assessing the relevance of developing methods for using gamification tools and gaming technologies in learning management systems", *Computer tools in education*, vol. 2, 2018, pp. 39–54.