

12/2 (2014), 213–228

tmcs@math.klte.hu
<http://tmcs.math.klte.hu>

Teaching
Mathematics and
Computer Science

Let’s Learn Database Programming in an Active Way

ANIKÓ VÁGNER

Abstract. This paper introduces how I applied the “learning by doing” method in the education of advanced knowledge of database systems in Software Engineering BSc program. The first goal of my method is to enable the students to use the PL/SQL and SQL as a skill, namely they get a practical competence which can be immediately used in business. In the laboratory the students independently practise the material learnt on the lecture. They get feedback for all their activities from the teacher. A software system helps administer the solutions, automatically verifies the syntax of them and helps the teacher to evaluate them. The paper summarises the results of three semesters. In the last year I compared the active learning method with the traditional method. I asked the students in a voluntary survey about the active learning method.

Key words and phrases: education, education of database systems, database, software aided education, education of programming, learning by doing, experimental learning.

ZDM Subject Classification: Q60.

Introduction

The main material of the Advanced DBMS 1 course in Software Engineering BSc program at the University of Debrecen, Faculty of Informatics is PL/SQL and advanced SQL in Oracle environment. The course consists of 2 hours lecture and 2 hours laboratory practice per week. In the lecture the students get acquainted

The publication was supported by the TAMOP-4.2.2.C-11/1/KONV-2012-0001 project. The project has been supported by the European Union, co-financed by the European Social Fund.

with the features of the Oracle database management system. In the laboratory they practise the new material which they have learnt in the lecture.

The lecture is traditional. The teacher shows the features to the students, and they take notes. In the lecture room there are no computers, they cannot try out the programs. The main goal of the laboratory is that the students try out the features of the Oracle. In the laboratory many teachers use the traditional educational method, which means that the teacher works at the whiteboard or projector and shows tasks with solutions to the students. With this method the students learn in a passive way, in most of the time they don’t ask, have no individual thoughts, they only sit and take notes. The teacher does not know whether they understand the solutions or not, whether they could solve the tasks alone or not. Sometimes the students do not give any feedback about their knowledge and problems. Maybe they try out the program codes at home, and practise at home. The first feedback is the result of the test paper. But this is too late for the students, because they don’t have enough time to correct the mistakes. The students most often do not ask questions about the problems, they learn something together that they think to be good. The classic didactic writers, like Skinner [12] and Polya [8] say that the learning process of students is more efficient if they learn in an active way instead of a passive one. The students should solve tasks and problems and acquire experience instead of observing the tasks and their solutions of the teacher. The tasks are to guide and help the students in the learning process. The “learning by doing” concept appear also in the new didactic books, like books of Schank, Berman & Macpherson [11] and Dufour, DuFour, Eaker & Many [2].

My idea is that I teach the Advanced DBMS 1 course in a way that I as a teacher give tasks to the students, so they practise SQL and PL/SQL alone, show the solutions to the teacher, and the teacher gives feedback to them. In this way the students learn in an active way.

I and one of my students created a software application framework to support the active learning method. This paper shows the active learning method, the software application framework, and evaluation of the method.

Literature Review

Although the new researchers often reject the old principle, the experimental learning is relevant nowadays. The concepts of Skinner [12] and Polya [8] can also be discovered in the recent didactic books. In their didactic book, Dufour,

DuFour, Eaker & Many [2] give an experimental education framework to the teacher. They show how the teachers can organize the work in the school and in the classroom. Roberts [10] presents a very good overview and historical survey of experimental education in his book.

The “learning by doing” concept works also in the education of computer science. Gogoulou, Gouli & Grigoriadou [4] used a software application for exploratory and collaborative learning in the education of programming. Drake [1] deals with experimental learning, but he points out that the active learning is not proper for every educational situation. In the area of database systems Ramakrishna [9] describes an experimental education survey of the undergraduate education. His results show that his students prefer the experimental learning over the traditional tutorials. Moore, Binkerd and Fant [7] describe a relational database management system course at Texas A & M University Corpus Christi that uses experimental learning. They receive a very good feedback from the participating students. Mason [6] also presents experimental learning for teaching database administration and software development at Regis University. His students indicated that the course was a successful experience that helps them fine-tune their technical skills and develop new soft skills.

Advanced DBMS 1 Course

The Advanced DBMS 1 course follows the material of the book titled PL/SQL Programming written in Hungarian [3], and uses also the actual documentation of Oracle [14]. The syllabus of the course can be found in the Appendix.

The prerequisite courses are Database Systems and Programming Languages 1 [13]. This means that the students who attend Advanced DBMS 1 course have learnt SQL and programming in C language. So the teacher of Advanced DBMS 1 can rely on the fact that the students can write SQL statements and programs in C.

The students get a signature at the end of the semester if they fulfil the requirements of the laboratory. If the students have a signature, they can take exam in the exam term. If they do not pass the exam, they can retake it 2 times. On the exam the teacher asks theoretical questions, like definitions of concepts, how the features (for example trigger, loop, cursor) work, what the proper syntax of a feature is, and so on.

The Active Learning Method

I have organized the work in the course so that it meets the following requirements:

- (1) In the laboratory the students work independently. I give tasks to the students in every lesson, and they solve it.
- (2) I give some feedback for every solution of each student.
- (3) If the solution is wrong, the students can give a new solution for the task before the deadline.
- (4) The students work on the tasks in the lessons, but if they do not finish the tasks, they can continue at home. If they complete the tasks in the lesson, they have no homework.
- (5) The tasks have deadlines. Because the students can work on the tasks at home, the deadline cannot be earlier than the end of the next lesson. If the students solve a task at home, they get the feedback on the next lesson. If the solution is wrong, they can give proper solution on the next lesson.
- (6) The students work independently, but they can discuss the tasks with each other.
- (7) The laboratory is based in the lecture; the students practise the new material of the lecture. In the laboratory there is no new material.
- (8) I as a teacher have to organize the lecture in a way that the students can work alone with the new material in the laboratory. The syllabus in the appendix shows that in the lecture there are a few topics which the students cannot practise or which can be practised without other topics. These topics are: types, character sets, lexical unit, locking, system triggers, etc. So, in the lecture the teacher should introduce materials which the students can practise in the laboratory.

Drake [1] finds problems with active learning, and he says that it can be used under given circumstances. In the case of Advanced DBMS 1 the student has previous knowledge about programming and database systems. They take part in the lectures, which show the new material and examples. So, first they get to know the material then they practise it. In the guidance the teacher has to find the golden mean: not too much but not too little. The first guidance is the lecture; the second are the tasks, which are getting more and more difficult during the lessons and the semester. The third guidance takes place in the laboratory, where some students need more explanation, some students need nothing. My

goal is to give feedback for all activities of the students; this means the laboratory activities and the homework (if there is). The interested students often write me emails with questions, and we have time to make conversations in the laboratory. The first goal of my active learning method is to enable the students to use the PL/SQL and SQL as a skill, namely they will get a practical competence which can be immediately used in business.

Laboratory Environment

The university has a standalone server on which Oracle Database 11g is installed. Every student has a personal database account. A schema belongs to each database account. In this schema the students can do everything they want. (Of course the database administrator gives privileges to students, but not all the privileges. So they can do everything they have privilege for.) In this schema, the students can explore how they can use the new material. The students can connect to the database from every computer which has Internet connection.

The students use SQL Developer as a client program. It is a very simple program, it needs no installation. The students copy it to the computer, and they can use it.

ADBMS Schema

I built a submit system to help the work in the laboratory. A similar system is used by Kósa, Pánovics, and Gunda [5] for contests. Their system evaluates the solutions, but it does not check how the competitor solved the tasks. My goal was to administer the tasks and solutions, check the syntax, but the semantics evaluation is done by the teacher.

The tasks and the solutions are also stored in the database. There is a schema named ADBMS, where tasks and solutions are stored. Figure 1 shows the tables of the ADBMS schema and the relationships between them.

The STUDENTS table stores the students who attend the course. The students attend this course in more than one section in a semester. Basically the schedules of the sections are the same in each semester, but it can occur that there is a difference between the numbers of the lessons because of public holidays. The SECTION attribute helps create groups of students for each section in a semester. The TASKS table stores the text of the tasks. It is worth dividing the tasks into groups because there are at least 3 tasks in one lesson, but this number can be 10

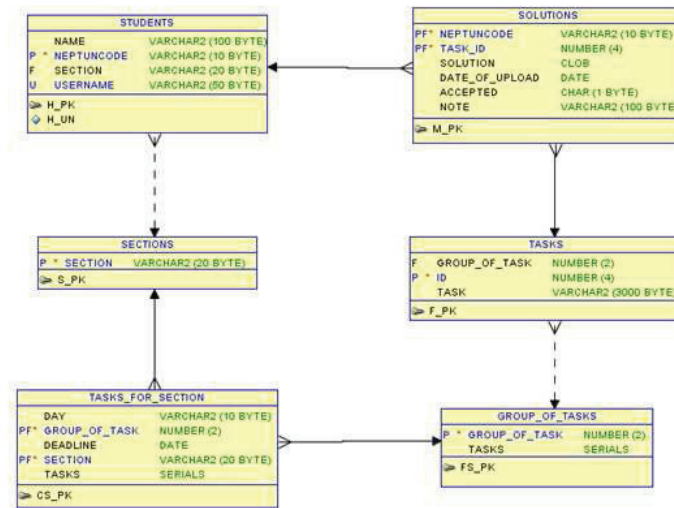


Figure 1. The ADBMS schema

or more. The GROUP_OF_TASKS table organises the tasks into groups. The TASKS_FOR_SECTION table assigns the tasks to the students. The students in each section get the same deadline for each group of tasks. The SOLUTION table stores the solutions of the students. It also stores the date when the student uploaded the solution, a flag which shows whether the solution is accepted by the teacher, and a note.

Each student can only read the data from the ADBMS schema, but not everything. This means that a student gets privilege only for selecting two views in the schema: MYTASKS and MYSOLUTIONS. The result of selecting the MYTASKS view contains the tasks (the identifier and the text of the task and the deadline of it) which the student has to solve. This is a subset of the rows in the TASKS table. The students can check their own personal uploaded solutions in the MYSOLUTIONS view. This view contains not only the accepted solutions but all uploaded solutions. The student can check in this view whether the solution of each task is accepted, has to be corrected, or is waiting for examination. In the view there is the comment column. The teacher writes comments into this column if the student does not participate in the lesson, and the solution of the task is not accepted. The comment shows the teacher and the student what the problem is with the solution and what has to be corrected.

A student can upload a solution into the database by executing the `UPLOAD_SOLUTION` procedure. The students have execution privilege for the `UPLOAD_SOLUTION` procedure. The procedure has two parameters. The first parameter is the identifier of the task; the second parameter is the text of the solution. The procedure automatically verifies the syntax of the solution. This means that the `UPLOAD_SOLUTION` procedure executes the uploaded solution in the schema of the student. If the program code cannot be executed in the schema, the solution will not be uploaded. The `UPLOAD_SOLUTION` procedure also checks the deadlines and throws an exception for the solutions where the deadline of the task has expired.

The Application for Supporting the Teacher

I and one of my students developed a software application to support the tasks of the teacher. The application has the following features:

- (1) Imports or inserts students and sections into the ADBMS schema. The data of the students can be changed with the application.
- (2) Publishes the text of the tasks. This means that the teacher writes the text of the new tasks and arranges the tasks into groups. Of course, later the text of the tasks can be modified. A task can be deleted until a student gives a solution to this task. The deadline of each group of tasks can also be determined. Of course the deadline can also be modified.
- (3) Helps the teacher verify the solutions which are uploaded by the students. This feature is the main reason why the application was developed. The program can list the text of the solutions with the name and the section of the student, the identifier of the task, and identifier of the group of tasks. The program also shows for each solution whether the solution is accepted, refused, or the teacher has not yet dealt with it. The program can filter and order the rows by these values. This list is very long at the end of the semester, so filtering is very important.

When a teacher chooses a solution from the list, the application shows the text of the solution, the text of the task, and the name of the student in another window. This window can be easily read. In this window the teacher can set whether the solution is accepted or refused. The teacher can also add a comment to the solution in this window. The teacher can do the semantic verification of the solution with the help of this window.

- (4) Lists the students who have not uploaded solutions for the given tasks. With this feature the teacher can easily find the students who are not ready with each task.
- (5) Lists the names of the students and shows how many per cent of all the tasks they have solved. The teacher can find the students who do not work hard enough in the semester or who have completed all the tasks.
- (6) Lists the name of students, and the groups of tasks and shows how many per cent of the tasks in the group are not solved by the student. With this function the teacher can check whether a student worked in the lessons or not.

Evaluation of the 2009/2010 Spring Semester

This was the first year when I used the active learning method. The requirements of the laboratory had two parts. One of them was that the students had to solve 75 per cent of the given tasks. The other part of the requirements was that the students had to write two test papers and they had to achieve 60 per cent of the maximum score of the two test papers. The students could use only a pen when they wrote the test paper; there was no computer and no documentation. If both requirements were fulfilled, the student got the signature.

In this year I did not give any deadline. A lot of students said that they would do the tasks the next week. At the end of the semester they wanted to finish the tasks. This was a hard job for them and also for me. The students gave solutions right before the last test papers, so I could not check in time who had enough solutions to write the test paper. So I promised that next year there would be deadlines, and that I would be strict in enforcing the deadlines.

The application was not ready, so I used SQL statements to verify the solutions. I spent a lot of time writing those SQL statements.

There was no syntactic verification in the `UPLOAD_SOLUTION` procedure. Some students uploaded solutions which did not work. And I did not find every syntactic problem in the solutions. These students did not want to try out the solutions.

There were 5 sections and 92 students. 48 students got signature. There were 2 students who solved all the tasks, gave proper solutions but could not achieve enough points on the test paper. The reason was that they copied the code from the documentation, and they did not remember the exact syntax without the

documentation. The other students did not give enough proper solutions in the laboratory.

Evaluation of the 2010/2011 Spring Semester

The requirements of the laboratory were the same as in the previous year. In this year I gave deadlines, and I was strict in enforcing them. The application was ready. I checked the solutions in the application, and I could check whether each student works or not. If the student did not work in the lesson, I could ask why. I could warn the students if they were behind with the solutions.

There was a syntactic verification in the `UPLOAD_SOLUTION` procedure. The students had more difficulties when they uploaded the solutions because it threw an exception if the solution could not be executed. My job was easier because I had to take care only of the semantics of the solutions.

There were 5 sections and 73 students. There were 15 students who solved all the tasks and gave proper solutions. They tried to be perfect. There were again 3 students who solved all the tasks, gave proper solutions but could not achieve enough points on the test paper. 51 students got signature. The other students did not give enough proper solutions in the laboratory or did not write the second test paper.

Evaluation of the 2011/2012 Spring Semester

In this year I wanted to compare the results of the traditional teaching method to the active learning method. There were 5 sections. In 2 sections I used the traditional teaching method; in the other 3 sections I used the active learning method.

The traditional teaching method means here that on the whiteboard I showed tasks and their solutions, which were related to the previous lecture. After the second week the students had to write a “small” test paper with one or two tasks every week. The first requirement of the laboratory was that the students had to give proper solutions for 60 per cent of the given tasks of the “small” test papers. There were 11 “small” test papers in the semester.

The active learning method was the same as in the previous years. The first requirement of the laboratory was that the students had to solve 80 per cent of the given tasks.

If the students fitted the first requirement, they could write the test papers at the end of the semester. They had to achieve 60 per cent of the maximum score of the test paper. The students could use only a pen when they wrote the test paper; there was no computer and no documentation.

In the sections where I used the traditional teaching method, there were 27 students. 10 students gave up the semester after a few “small” test papers. This means that they wrote 5, 6, or 7 “small” test papers, and they did not attend the course any more. 14 students wrote the final test paper. 13 students got signature at the end of the semester. So I can say that the students who were learning continuously in the semester wrote successful “small” test papers, and the final test paper was easy for them.

In the 3 sections where I used the active learning method, there were 36 students. 4 students could not solve 80 per cent of the given tasks. They were not allowed to write the final test paper. 21 students got signature at the end of the semester.

Comparing the Two Teaching Methods Based on the Performance of the Students

On the first lesson 60 students wrote a test paper about their practical knowledge of programming and database systems. They had to write a C program code and SQL statements. Table 1 shows the results of this test paper.

Table 1. Results of the test paper of the first lesson

	TT students	AL students
Number of students	26	34
Got signature	50%	61%
Average result of programming part	50.64%	51.96%
Average result of SQL part	25%	23.39%

The results of the test paper were similar in both cases. We can see from the table that the AL (active learning) students got more signatures than the TT (traditional teaching) students. Based on that table I can say that the students who learnt with the active learning method performed better during the course.

Table 2 shows the average of the exam mark of the students. 31 students took the exam in this semester. There is no significant difference between the

results of the two groups. Based on Table 2 we can say that the exam mark did not depend on the teaching method of the laboratory.

Table 2. The average of the exam marks of the students

	TT students	AL students
Number of students	11	20
Average of exam marks	2.54	2.45

On the exam the teacher asks theoretical questions instead of practical tasks. This way the better practical knowledge does not influence the exam mark. The students who are not so good in laboratory can mug the concepts and other theoretical questions and get a good mark. On the other hand, the students who are very good in practice may not learn everything for the exam.

Common Observations Regarding the Active Learning Method

There were students who copied the solutions of other students. They did nothing but tried to pass the course without any work. When I recognized it, I told to the students that I did not mind it. But they had to learn because the students had to work on the test papers individually, and they can learn by solving the tasks.

There were students who started to solve the tasks the day before the deadline, namely, their first upload was less than 24 hours before the deadline. This means that they had no time to correct their wrong solutions. I tried to motivate them, but there was no result. In most cases they did not get a signature. In the 2011/2012 spring semester there were 10 students who tried to solve more than half of the tasks late and did not get a signature. This number was 9 in the 2010/2011 spring semester.

There was only one student in the 2011/2012 spring semester who solved all the tasks without being late. He gave proper solutions for 98% of the tasks and got excellent (5) on the exam.

There were students who wanted to solve every task with a proper solution. They wanted to learn everything. In the 2011/2012 spring semester there were 3 such students, 2 of them got the best mark (5) at the exam. In the 2010/2011 spring semester there were 15 students who gave proper solutions for 100% of

the tasks. On the exam they got various marks. I cannot say that they learnt everything.

The result of the exam does not depend on the result of the laboratory. There were students who solved everything in the laboratory and got only sufficient mark (2) or fail (1) on the exam. The students who were not so good in the laboratory could be good (4) or excellent (5) on the exam. Generally I can say that if the students got a signature, they could get the sufficient mark easily. For a better mark they had to learn more; what they learnt in the laboratory was not enough.

Voluntary Survey of the Students

I asked the AL students about the active learning method. I got 63 answers. 71% of the respondents passed the exam. 90% of the respondents liked that they had to solve the tasks individually in the laboratory. 80% of the respondents would not have liked that the teacher solved the tasks in the laboratory. 14% of the respondent did not mind it. 54% of the respondents would like the active learning method to be applied on other courses. For 19% of the respondents it makes no difference. 75% would choose the active learning method in the laboratory of the Advanced DBMS 1 instead of the traditional style. 12% of the respondents do not mind it. 7% of the respondents said that they uploaded solutions without trying it out.

The students gave other individual feedbacks. Here are the most important opinions:

- It was a good experience to solve the tasks alone.
- I could not solve the tasks week by week.
- I liked this teaching method because I had to think about the solutions of the tasks. I did not need to follow the thinking of the teacher.
- Every student could work in an individual pace.
- Sometimes there were a lot of tasks for a lesson. I had to solve the tasks at home, and I got feedback only on the next lessons. So I had to work with the tasks of previous lessons instead of the new tasks.
- I understood the tasks better thanks to the individual work.
- I was forced to understand the usage of the material in the laboratory. I think that the active learning method should be used at most courses. The system was strict, but I did not feel that it goes against the students.

- Consistent, precise, and strict requirements.
- I would have liked if the teacher had given more solutions and comments, and we had been thinking together. I am bored of working alone.
- I did not attend the lectures, so I would have liked to get a short summary at the beginning of the laboratory.
- The active learning method was very good in improving programming skills and solving problems.
- I liked that I could work alone, and if I had a problem, I received assistance.
- The students were forced to solve the tasks. It is not enough to listen to how they can be solved.
- I liked the active learning method. I got answers to all of my questions.
- The students had to discover how the features work. I understood the solutions better, and I remember the material better. I got help if I had problems.

I asked the TT students about the active learning method. I got 10 answers. 50% of the respondents would have liked to attend the section applying the active learning method. A student liked when I gave more than one solution for a task.

This result is similar to the results of Ramakrishna [9] and Mason [6].

Conclusion

The “learning by doing” or active learning method is widely used and working properly also nowadays. In the education of programming it is very important that the students practise independently. The active learning method forces them to write program code and use the database management system independently. The task of the teacher is to guide and help the students through the material. The teacher gives personalized answers to the students, motivates them, and discusses the problems and the solutions. The results of the students show us that the laboratory results are definitely better if the teacher uses the active learning method. The results of the voluntary survey show us that students liked the active learning method; moreover, they would like it at other subjects too. The students have to work hard during the semester, but at the end of the semester, they have experience in using the database management system, and they will not forget it for many years.

References

- [1] J. R. Drake, A Critical Analysis of Active Learning and an Alternative Pedagogical Framework for Introductory Information Systems Courses, *Journal of Information Technology Education: Innovations in Practice* **11** (2012), 39–52.
- [2] R. Dufour, R. DuFour, R. Eaker and T. Many, *Learning by Doing: A Handbook for Professional Communities at Work*, United States of America: Solution Tree Press, 2010.
- [3] A. Gábor and I. Juhász, *PL/SQL programozás [PL/SQL Programming]*, Budapest: Panem, 2007.
- [4] A. Gogoulou, E. Gouli and M. Grigoriadou, Teaching programming with ECLiP didactical approach, International Conference on Cognition and Exploratory Learning in Digital Age, 2009, 204–211.
- [5] M. Kósa, J. Pánovics and L. Gunda, An Evaluating Tool for Programming Contests, Vol. I, 6th International Conference on Applied Informatics, 2004, 163–172.
- [6] R. T. Mason, A Database Practicum for Teaching Database Administration and Software Development at Regis University, *Journal of Information Technology Education: Innovations in Practice* **12** (2013), 159–168.
- [7] M. Moore, C. Binkerd and S. Fant, Teaching web-based database application development: an inexpensive approach, *United States of America: Journal of Computing Sciences in Colleges* (2002), 58–63.
- [8] G. Polya, *How to solve it*, United States of America: Princeton University Press, 1957.
- [9] M. V. Ramakrishna, A Learning by Doing Model for Teaching Advanced Databases, *Proceedings of the Australasian conference on Computing education* (2000), 203–207.
- [10] J. W. Roberts, *Beyond Learning by Doing*, New York: Routledge, 2011.
- [11] R. G. Schank, T. R. Berman and K. A. Macpherson, Learning by Doing, *Instructional Design Theories and Models*, United States of America: Lawrence Erlbaum Associates, 1999, 161–182.
- [12] B. F. Skinner, *The technology of teaching*, New York: Meredith Corporation, 1968.
- [13] Bulletin of Software Engineering BSc program at the University of Debrecen, 2007, Retrieved from http://www.inf.unideb.hu/oktatas/?cat=&site=hallgato/nappali/oklevel_kovetelmeny/pti_2007
English version retrieved from http://www.englishstudies.unideb.hu/images/documents/SoftwareIT_BSc_2011.pdf.
- [14] Oracle Documentation, 2013, Retrieved from <http://www.oracle.com/pls/db112/homepage>.

Appendix

Syllabus of Advanced DBMS 1 lecture

- (1) Main features of PL/SQL, PL/SQL block, constant and variable declaration, basic types of PL/SQL, DBMS_OUTPUT.PUT_LINE procedure, statements
- (2) Data types of PL/SQL, SELECT INTO statement, SQL DML with RETURNING
- (3) Character set, lexical units, expressions in PL/SQL
- (4) Subprograms in PL/SQL
- (5) Exceptions and exception handling in PL/SQL, CURSOR FOR LOOP statement
- (6) DML triggers
- (7) System triggers, updatable and non-updatable views
- (8) Cursors and cursor variables
- (9) Transaction processing and control in PL/SQL, Locking in PL/SQL
- (10) Packages
- (11) Composite types of PL/SQL 1.
- (12) Composite types of PL/SQL 2.
- (13) Native dynamic SQL
- (14) Write efficient PL/SQL program
- (15) Other topic of PL/SQL programming

Syllabus of Advanced DBMS 1 laboratory practice

Tasks of each lesson can ask to use the programming tools of previous lesson.

- (1) Practise of SQL
- (2) Tasks using blocks, declarations, statements, and the DBMS_OUTPUT.PUT_LINE procedure
- (3) Tasks about data types of PL/SQL, SELECT INTO statement, SQL DML with RETURNING
- (4) Tasks of expressions in PL/SQL

- (5) Tasks of Subprograms in PL/SQL
- (6) Tasks of exceptions and exception handling in PL/SQL, CURSOR FOR LOOP statement
- (7) Tasks of DML triggers
- (8) Tasks of updatable and non-updatable views
- (9) Tasks of cursors
- (10) Tasks of cursor variables
- (11) Tasks of packages
- (12) Tasks of composite types of PL/SQL 1.
- (13) Tasks of composite types of PL/SQL 2.
- (14) Tasks of native dynamic SQL
- (15) Writing examination paper done under supervision

ANIKÓ VÁGNER
FACULTY OF INFORMATICS
UNIVERSITY OF DEBRECEN
4028 DEBRECEN, KASSAI U. 26.
HUNGARY

E-mail: `vagner.aniko@inf.unideb.hu`

(Received January, 2014)