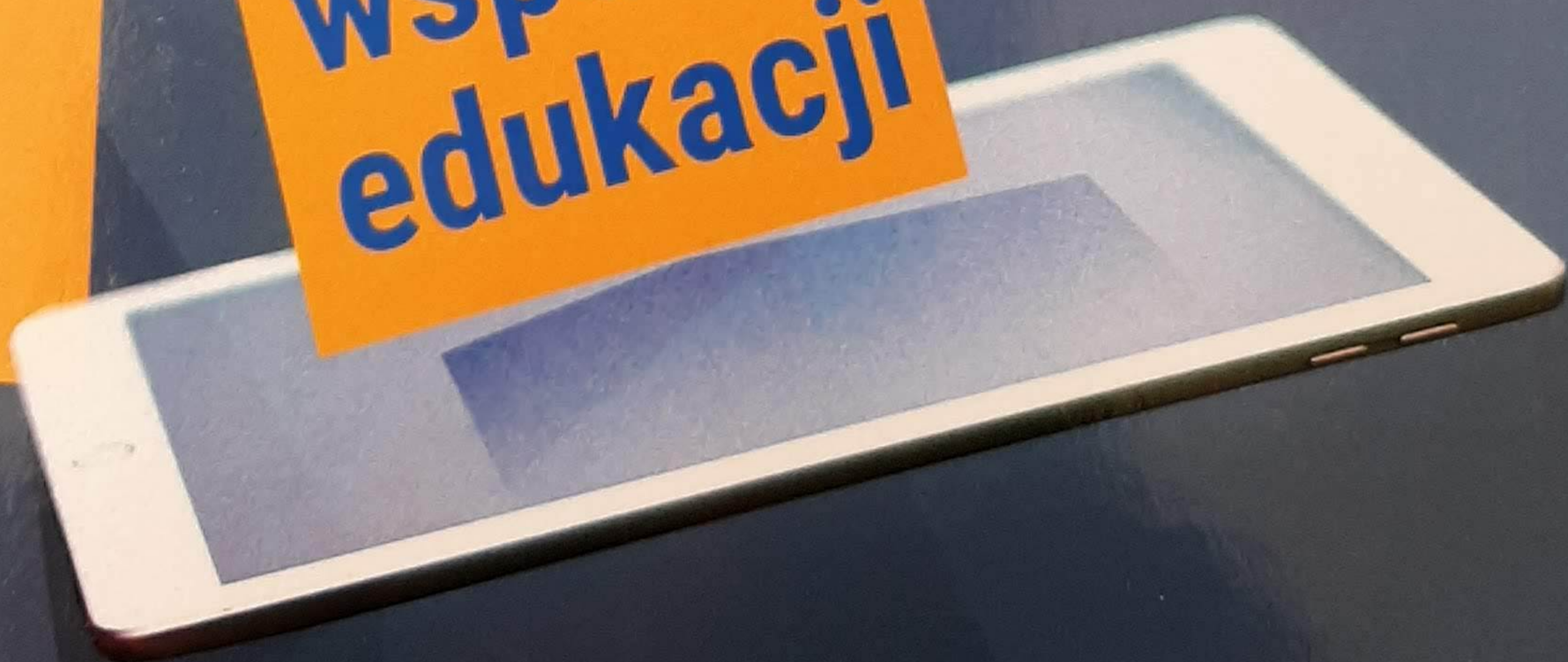




UNIWERSYTET
RADOMSKI

Pod redakcją naukową
Elżbiety Sałaty i Katarzyny Ziembakowskiej-Cecot

**Aktualne
wyzwania
współczesnej
edukacji**



Recenzenci:

doc. Ing. Melánia Feszterová, PhD. Univerzita Konštantína Filozofa v Nitre
dr Krzysztof Symela, Instytut Technologii Eksploatacji

Opracowanie wydawnicze
Monika Fetráš

Projekt strony tytułowej
Mariusz Dański

Copyright © by Andor Abonyi-Tóth, Imre Bende, Maciej Ciepiela, Pavel Dostál, Oľga Dúbravská, Bianka Francistyová, Silvia Hajníková, Bálint Horcsin, Maksym Horinshtein, Roman Horváth, Juliána Litecká, Zuzana Mitaľová, Oksana Nagorniuk, Władimir Pasiecznik, Milan Pokorný, Kacper Prauzner, Tomasz Prauzner, Ildikó Pšenáková, Ladislav Rudolf, Volodymyr She, Svatopluk Slovák, Eugeniusz J. Sobczyk, Wiktoria Sobczyk, Radim Štěpánek, Ján Stoffa, Veronika Stoffová, Veronika Švrčinová, Karolina Szczęsna, Václav Tvarůžka, Jan Vaněk, Ján Višňovský, Maria Vysochanska, Anastazja Wilczkowska

Copyright © by Uniwersytet Radomski
im. Kazimierza Pułaskiego (Wydawnictwo 2023)
26-600 Radom, ul. Malczewskiego 29
www.uniwersytetradom.pl, e-mail:wydawnictwo@uthrad.pl

ISSN 1642-5278

ISBN 978-83-7351-991-6

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości bądź fragmentów niniejszej publikacji w jakiejkolwiek postaci jest zabronione. Wykonywanie kopii metodami: kserograficzną, fotograficzną, a także kopiowanie na nośnikach: filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich do niniejszej publikacji.

Autorzy dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności za ich wykorzystanie ani za związane z tym ewentualne naruszenie praw autorskich lub patentowych. Autorzy nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Wyd. 1

Contents

Introduction.....	7
-------------------	---

CURRENT PROBLEMS OF CONTEMPORARY EDUCATION

Milan Pokorný ON STUDENTS' OPINIONS ON ONLINE LEARNING AND BLENDED LEARNING OF MATHEMATICS.....	11
Ildikó Pšenáková UTILIZATION OF ONLINE EDUCATION IN INFORMATICS SUBJECTS	21
Veronika Stoffová, Ján Stoffa, Silvia Hajníková SOME TERMINOLOGICAL PROBLEMS OF USING MODERN INFORMATION AND COMMUNICATION TECHNOLOGIES.....	31
Włodimir Pasiecznik, Anastazja Wilczkowska FUNCTIONING OF EDUCATION IN UKRAINE UNDER EXTREME CHALLENGES	40
Oksana Nagorniuk, Maciej Ciepiela, Eugeniusz J. Sobczyk, Wiktoria Sobczyk THE ROLE OF EDUCATION DURING THE WAR. EDUCATIONAL AND HUMANITARIAN AID FOR REFUGEES FROM UKRAINE	51
Maria Vysochanska, Volodymyr She, Maksym Horinshtein, Wiktoria Sobczyk ORGANIZATION OF DISTANCE STUDY AND ONLINE EDUCATION IN TIMES OF CRISIS AND WAR.....	61
Ján Višňovský, Bianka Francistyová, Oľga Dúbravská ARTIFICIAL INTELLIGENCE – RISKS AND OPPORTUNITIES IN HIGHER EDUCATION IN MEDIA AND COMMUNICATION STUDIES.....	71
Radim Štěpánek, Svatopluk Slovák, Jan Vaněk, Pavel Dostál, Veronika Švrčinová FINANCIAL EDUCATION AT PRIMARY SCHOOLS IN THE OSTRAVA DISTRICT	81
Karolina Szczesna SPEECH LEVEL OF SIX-YEAR-OLD CHILDREN.....	90

SELECTED IT AND MATHEMATICAL APPLICATIONS IN EDUCATION

Veronika Stoffová APPLICATION OF ARTIFICIAL INTELLIGENCE IN EDUCATION	105
Juliána Litecká, Zuzana Mitaľová THE DIDACTICAL USING OF 3D PRINTING IN TECHNOLOGY SUBJECT TEACHING	116
Bálint Horcsin, Andor Abonyi-Tóth NOVEL DATABASE MANAGER AGNOSTIC METHOD FOR AUTOMATICALLY CHECKING THE CORRECTNESS OF A SUBMISSION FOR A DATABASE MANAGEMENT TASK	126
Imre Bende EFFICIENCY TEST OF AN ALGORITHM VISUALIZATION TOOL	142
Roman Horváth REPORT ON THE IMPLEMENTATION OF THE ADDITIONAL SCREEN COMPONENTS OF THE GROBOT PROGRAMMING FRAMEWORK	149
Tomasz Prauzner, Kacper Prauzner INTERPRETATION OF QEEG AND EYE TRACKING TESTS IN THE ASSESSMENT OF TEACHING ACTIVITY	157
Ladislav Rudolf, Václav Tvarůžka THE USE OF CORRELATION ANALYSIS IN EDUCATIONAL RESEARCH	168
Václav Tvarůžka, Ladislav Rudolf GEOMETRIC BODIES AS A TOPIC OF TECHNOLOGY AND TECHNICAL PRINCIPLES TEACHING	174
Notes about authors	182

Bálint Horesin
ORCID: 0009-0003-7222-1013
Andor Abonyi-Tóth PhD.
ORCID: 0000-0001-7629-0233

NOVEL DATABASE MANAGER AGNOSTIC METHOD FOR AUTOMATICALLY CHECKING THE CORRECTNESS OF A SUBMISSION FOR A DATABASE MANAGEMENT TASK

INTRODUCTION

There have been plans on our faculty to create a system that can help students in learning for the “*Databases I.*” course and teachers checking their tests. Even though some plans have been made, unfortunately none of the projects have reached the phase of actual development.

While there have been some important milestones reached by our faculty with the Bíró / Mester systems which have been helping numerous students learning and practicing programming for many, it should be noted that the success of those systems come from being used by students on the faculty and high-school students preparing for the school leaving examination and also for various national and international competitions (Horváth, Horváth, & Zsakó, 2017).

So, we deemed it necessary to help both the secondary school system of Hungary and also make it usable on the faculty.

But while universal programming languages usually have standardized input and output, where such systems usually communicate in some way, database management systems do not come with such features. Therefore, a new and abstract way of communicating with the database management systems was necessary to be established so that creation of tasks would be possible in a reasonable timeframe.

The research presented in this paper has created a novel method that allows task creation to be done easily.

This paper contains some information on the functioning of the secondary school system of Hungary to give the background necessary to understand why certain goals have been set, what achievements previous research has made, what is our method for checking SQL queries, demonstrate its functions, describe how it compares to previous research, and how it could be improved in the future.

THE TWO-LEVEL SCHOOL LEAVING EXAM IN HUNGARY

The two-level school leaving examination was established in 2005 in Hungary. In this system when students finish the 12th, they undertake an examination that is common in the entire world.¹ Students can choose between intermediate and advanced levels for each subject. The advanced level exams have replaced the entrance examination for higher education. For this reason, it is not possible to take an advanced level examination in all of the subjects, only in those that are relevant to the higher education admission process.

A student can only acquire a school leaving examination certificate if they have successfully passed the exams in at least five subjects. Four of these subjects are required by law (Hungarian language and literature, mathematics, history and a foreign language), the fifth one is chosen by the examinee. The exams in most subjects consist of a written and an oral section.

Oral exams are always evaluated by a committee. The chairman is appointed by the Educational Authority and the other committee members are assigned by the director of the school on the intermediate level. On advanced level exams the committee is assigned by the Educational Authority in such a way that committee members are not the teachers of the examinees.

In the case subjects where the requirements are set centrally, the question papers and their scoring guidelines used in the written examinations are prepared by a committee. The committee members are invited by the Educational Authority. There are multiple prepared question papers and one of them is selected randomly for the given exam period by the Educational Authority. The Department produces the required number of papers and forwards them to the locations of the exams.

The written examinations are evaluated in accordance with the scoring guidelines by a qualified teacher. The teacher is assigned by the director of school on the intermediate level, and the teacher is chosen by the Government Office on the advanced level similar to the selection of the committee for the oral part.

Exam papers from previous years can be accessed on the website of the Educational Authority.

¹ It is called Matura examination in multiple countries (including Hungary), but we will reference it as school leaving examination for the readers to have a better understanding. We will use the phrase "student" for students learning in secondary schools (high-schools) which might not be usual for some readers.

Bálint Horcsin
ORCID: 0009-0003-7222-1013
Andor Abonyi-Tóth PhD.
ORCID: 0000-0001-7629-0233

NOVEL DATABASE MANAGER AGNOSTIC METHOD FOR AUTOMATICALLY CHECKING THE CORRECTNESS OF A SUBMISSION FOR A DATABASE MANAGEMENT TASK

INTRODUCTION

There have been plans on our faculty to create a system that can help students in learning for the “*Databases I.*” course and teachers checking their tests. Even though some plans have been made, unfortunately none of the projects have reached the phase of actual development.

While there have been some important milestones reached by our faculty with the Bíró / Mester systems which have been helping numerous students learning and practicing programming for many, it should be noted that the success of those systems come from being used by students on the faculty and high-school students preparing for the school leaving examination and also for various national and international competitions (Horváth, Horváth, & Zsakó, 2017).

So, we deemed it necessary to help both the secondary school system of Hungary and also make it usable on the faculty.

But while universal programming languages usually have standardized input and output, where such systems usually communicate in some way, database management systems do not come with such features. Therefore, a new and abstract way of communicating with the database management systems was necessary to be established so that creation of tasks would be possible in a reasonable timeframe.

The research presented in this paper has created a novel method that allows task creation to be done easily.
This paper contains some information on the functioning of the secondary school system of Hungary to give the background necessary to understand why certain goals have been set, what achievements previous research has made, what is our method for checking SQL queries, demonstrate its functions, describe how it compares to previous research, and how it could be improved in the future.

THE TWO-LEVEL SCHOOL LEAVING EXAM IN HUNGARY

The two-level school leaving examination was established in 2005 in Hungary. In this system when students finish the 12th, they undertake an examination that is common in the entire world.¹ Students can choose between intermediate and advanced levels for each subject. The advanced level exams have replaced the entrance examination for higher education. For this reason, it is not possible to take an advanced level examination in all of the subjects, only in those that are relevant to the higher education admission process.

A student can only acquire a school leaving examination certificate if they have successfully passed the exams in at least five subjects. Four of these subjects are required by law (Hungarian language and literature, mathematics, history and a foreign language), the fifth one is chosen by the examinee. The exams in most subjects consist of a written and an oral section. Oral exams are always evaluated by a committee. The chairman is appointed by the Educational Authority and the other committee members are assigned by the director of the school on the intermediate level. On advanced level exams the committee is assigned by the Educational Authority in such a way that committee members are not the teachers of the examinees.

In the case subjects where the requirements are set centrally, the question papers and their scoring guidelines used in the written examinations are prepared by a committee. The committee members are invited by the Educational Authority. There are multiple prepared question papers and one of them is selected randomly for the given exam period by the Educational Authority. The Department produces the required number of papers and forwards them to the locations of the exams.

The written examinations are evaluated in accordance with the scoring guidelines by a qualified teacher. The teacher is assigned by the director of school on the intermediate level, and the teacher is chosen by the Government Office on the advanced level similar to the selection of the committee for the oral part.

Exam papers from previous years can be accessed on the website of the Educational Authority.

¹ It is called Matura examination in multiple countries (including Hungary), but we will reference it as school leaving examination for the readers to have a better understanding. We will use the phrase “student” for students learning in secondary schools (high-schools) which might not be usual for some readers.

REQUIREMENTS OF THE INFORMATION TECHNOLOGY SUBJECT

The first National Curriculum has been published in 1995 consisting of 10 areas that every Hungarian student needed to learn. One of those was information technology. This curriculum came to effect in 1997 and by that, information technology as a separate subject has been introduced in every school in the country.

Information technology has both a written and an oral section in the school leaving examination since the introduction of the two-level school leaving examination in 2005. On the intermediate level the written part is 180 minutes, the oral part is 15 minutes long. On the advanced level exam pupils have 240 minutes to solve the written part, while the oral part is 20 minutes long.

The written part is executed on machines without an internet connection. Its topics, maximal scores and recommended time periods are described in the table 1.

Table 1. Basic information about information technology subject's school leaving examination

Information technology subject – intermediate level written exam		
Topic	Recommended time frame	Maximal score
Word processing	60 minutes	40 points
Creation of presentations, graphics, websites	40 minutes	30 points
Spreadsheet management	50 minutes	30 points
Database-management	30 minutes	20 points
Information technology subject – advanced level written exam		
Topic	Recommended time frame	Maximal score
Word processing, creation of presentations, graphics, websites	60 minutes	30 points
Spreadsheet management	30 minutes	15 points
Database-management	60 minutes	30 points
Algorithms, programming	90 minutes	45 points

The biggest difference between the topics of the intermediate level and the advanced level is that tasks about algorithms and programming are only present on the advanced level (While for example database management are in both levels.). The tasks on the advanced level are generally harder since they require a broader knowledge in all of the topics.

On the advanced level examinations, the list of available software is described by the Educational Authority to provide uniform conditions for each examinee. Sometimes pupils have the option to choose the software they want to use to solve the tasks. For example, in the autumn of 2023, they have the option to use Microsoft Access 2019 or LibreOffice Base 7.3 (with MySQL 8.0 database engine) software to solve the database-management task on Windows 10. While if they choose to use

Linux, they need to use LibreOffice 6.4.6 Base with MySQL 8.0 for the database-management task.

On the oral part, they get a topic randomly from at least 20 different topics and they have 30 minutes to prepare for their examination conducted by a committee.

THE INTRODUCTION OF THE DIGITAL CULTURE SUBJECT AND ITS REQUIREMENTS

A new National Curriculum has been introduced in 2020 in Hungary starting with the first, fifth and ninth classes. The material of the information technology subject has been drastically changed. The weekly number of lessons has been increased, and its name has also been changed to "Digital culture". The requirements of the school leaving examination have also been changed.

If a pupil gets at least 25% on the written parts, they are not needed to take an oral part. Taking an oral exam is only needed if they score at least 12% but do not reach 25%.

The written part is executed on machines without an internet connection. Its topics, maximal scores and recommended time periods are described in the table 2.

Table 2. Basic information about digital culture subject's school leaving examination

Digital culture subject – intermediate level written exam		
Topic	Recommended time frame	Maximal score
Word processing	45 minutes	25 points
Visual elements (graphics, image manipulation, creation of presentations)	35 minutes	20 points
Spreadsheet management	40 minutes	25 points
Database-management	30 minutes	15 points
Algorithms and programming	30 minutes	15 points
Digital culture subject – advanced level written exam		
Topic	Recommended time frame	Maximal score
Creation of documents (word processing, graphics, image manipulation, creation of presentations, website creation) OR Spreadsheet management ²	70 minutes	35 points
Database-management	70 minutes	35 points
Algorithms, programming	100 minutes	50 points

² The examinee can choose if they want to solve the task of creating the described document or if they want to solve the task about spreadsheet.

On the intermediate level it is recommended for schools to provide either LibreOffice Base or Microsoft Access on the intermediate level. But it is mandatory to use MariaDB with PHPMyAdmin on the advanced level.

On the advanced level oral exam, the examinee may use a computer without internet access, and use a computer with internet access during their examination for demonstrational purposes. There are two parts of the oral examination:

Task: describe a topic by addressing certain described criteria from word processing, graphics, image manipulation, creation of presentations, spreadsheet or database-management which is selected randomly.

Task: solving and presenting one or more programming tasks. If the task requires a coding, editing of code or interpretation of code, the examinees might use a computer without internet access during their preparation.

SUPPORTED DATABASE-MANAGEMENT SYSTEMS

As described above, during the introduction of the digital culture subject, many aspects have changed of how information technology is taught in the schools in Hungary.

We are using Oracle Database to teach most of our students database-management and use PostgreSQL to teach students that are on the "evening curriculum" (their subjects and timetables are a bit different than the students' that are learning normally during the day). We also wanted to support pupils in secondary education, therefore we also needed to add support for MariaDB, LibreOffice (with MySQL database engine) and Microsoft Access.

Developing different methods for all 5 of these softwares was not feasible. Therefore, a new way of creating tasks was required that allowed the creation of a task in a database-agnostic way. Some implementation and testing are needed for each software but once a database-management system is supported, all previous and upcoming tasks would be supported by it. While succeeding with this criterion, we also wanted it to be able to check for tasks that appeared on the school leaving exams.

Currently, our software only supports Oracle Database, MariaDB, MySQL database engine and Microsoft Access. We are planning to also make the implementation this autumn for PostgreSQL.

GOALS

Our main goal was to research the methodology of a platform that would be available for high-school students and students in our faculty that would help them with learning the basics of a structured query language chosen by them from the aforementioned list.

The platform will have some written materials and tasks that help students practice the skills in database-management. This paper focuses on how the grading of such tasks could be done automatically.

The targeted user group have varying levels of prior knowledge (some might have not even learned about database-management, some might be self-taught, some

might just want to prepare for their university exams), their solutions could be correct and could reach maximal score on an examination while being non-idiomatic³. Since one of our platform's main target audiences is students around the country learning database-management on their own, we concluded that giving negative feedback to students whose code is correct but non-idiomatic would be demotivating and might dissuade them from using our platform or even quit learning database-management. It has been concluded in meta-analysis (Fong, Patall, Vasquez, & Stautberg, 2018) that giving negative feedback is less demotivating if the student gets instructions on how they could improve, understands the criteria and get the feedback in-person. None of this can be achieved with an online platform be guaranteed by an automated system, since giving correct instructions cannot on the school leaving exam and does not have a clear and understandable definition. Providing feedback in-person is understandably impossible.

Therefore, we have concluded that the platform always needs to have a relatively high accuracy, the results should not be influenced by the quality of code written by the students and should almost never give negative feedback if a student submits code that gives correct result. Due to the limitations described by Rice's theorem (some of these languages are Turing-complete so their correctness cannot be decided using any known machine and if the Church-Turing thesis is correct, there will never be such machine), we needed to establish the constraint that the submitted code should not be so slow that a computer cannot evaluate it in a reasonable time limit. It should be noted that additional research is needed to determine if it is possible to check the performance of a SQL script with great accuracy.

The query types that we want to be able to check are from data query language (SELECT), data manipulation language (INSERT, UPDATE, DELETE), data definition language (CREATE, DROP, ALTER, TRUNCATE)⁴. This is in accordance with the requirements of the school leaving examination both in the subject of information technology and on the subject of digital culture.

PREVIOUS SOLUTIONS

There are mainly two ways of checking the correctness of a computer program. One of them is black-box testing where it is checked how the software reacts

³ I. Bálint Horesin – one of the authors of this paper – had used extensive amounts subqueries in my school leaving examination since I have mostly self-taught myself in this topic.

⁴ Structured Query Language (SQL) is a declarative language widely considered as consisting of 5 sublanguages: data query language (DQL) which has the SELECT statement, data manipulation language (DML) which can change rows of tables, data definition language (DDL) which can change the schema (structure of tables), transaction control language (TCL) which can control database transactions, data control language (DCL) which makes identity and access management function on database-management systems.

in certain situations and check if that reaction was appropriate. The other way is conducting code analysis in order to perform white-box testing.

It has been correctly stated in (Dekeyser, de Rudder, & ...)

It has been correctly stated in (Dekeyser, de Raadt, & Lee) that relational query languages are not Turing-complete, but most of the languages we needed to be supported have certain features that allows them to be Turing-complete. By creating a special flavor of SQL it is possible to give not only quantitative (correct / incorrect) but also qualitative (how a student could improve) feedback as it is shown in (Shah & Pareek, 2022), but the National Core Curriculum would need to be changed beforehand which was outside the scope of our research. There has been research done on the University of Szeged which is in (Korom). It checks if that reaction was appropriate. The other way

There has been research done on the University of Szeged which is described in (Korom). It checks how close that submitted code is how close to the correct solution(s).

To reach the goals mentioned above, we have decided that black-box testing is required since the quality of code should not interfere with the results. Black-box testing has been successfully used in multiple systems helping students learn, practice, have exams and competitions. While this approach has been debated by multiple authors, including (Dekeyser, de Raadt, & Lee), (Shah & Pareek, 2022) and (Ribeiro & Guerreiro, 2009), it is still used on multiple events, like on the International Olympiad in Informatics, in Hungarian national competitions and many online platforms (the likes of Codeforces, CodeChef, SPOJ, Bíró, Mester, Progcot). There has been various research discussing black-box testing for

There has been various research discussing black-box testing for database management systems. We will mostly discuss the solutions described by (Király, Balla, & Király, 2022) and (Kádek, Kósa, Orosz, & Tóth, 2017). Their solutions work with MySQL and uses a shared database-management system.

OUR METHOD

Our platform uses a web interface where students can submit their solutions. If there is enough processing power available, the evaluation of the submission begins. If enough resources are not readily available, then it is put into a queue. It is common practice for such systems, like the software available at <https://cms-dev.github.io/>.

It is common practice while evaluating universal programming languages to create some kind of sandbox where the program being executed has limited access to software resources. It is normally done using some capabilities of the Linux kernel, the specifications of such an implementation is discussed by the documentation of software available at <https://github.com/loi/isolate>.

During practice the system could give the user their output and the correct output which – with small test cases – could be used as an easily understandable way of showing what is the problem with the query.

The solutions discussed in (Király, Balla, & Király, 2022) and (Kádek, Kósa, Orosz, & Tóth, 2017) used a shared database process during the black-box testing. In (Kádek, Kósa, Orosz, & Tóth, 2017) they used users with the permission only to `SELECT` when testing `SELECT` queries and used transactions to check other types of queries. In (Király, Balla, & Király, 2022) they used temporary tables during testing.

Giving any kind of permission on a database-management software to anybody having internet is not common practice. And we have also realized that using shared databases would inherently cause the entire system to be vulnerable to unique security holes from each supported database-management system. Therefore, each submission gets its own sandbox which is created where it gets database-management system. Limiting the resources of such sandboxes is a documented task. One could use Docker to create a sandbox for some but for Microsoft Access it is easier to create a virtual machine (e.g. using VMWare). While it is an interesting topic, how it can be done is beyond the scope of this paper.

Giving any kind of permission to a user having internet is not common practice. And databases would inherently cause the entire system to have security holes from each supported database-management system. Therefore, each submission gets its own sandbox which is created where it gets its own database-management task. One could use Docker to create a sandbox for some of these programs, but for Microsoft Access it is easier to create a virtual machine (with something like Vagrant). While it is an interesting topic, how it can be done more efficiently, it is well outside the scope of this paper.

It also comes with the benefit that any kind of script can be tested in the future, from the likes of data control language (GRANT, REVOKE) and transaction control language (COMMIT, ROLLBACK). So we can also give users SYSDBA / root access to the database without introducing security vulnerabilities.

Once a sandbox has been created, the appropriate database-management software for the submission, and the software that controls the evaluation is started. Once a sandbox has been created, the appropriate database-management software for the submission, and the software that controls the evaluation is started.

It also comes with the ability to roll back transactions (COMMIT, ROLLBACK). So we can undo changes made from the likes of data control language (DDL) statements. This software connects to the database and executes the black-box testing which's exact method will be discussed later. We have written the software in C# on .NET Framework because properly supported connection libraries exist for all the required database management systems (Microsoft Access, MySQL (for LibreOffice), MariaDB, PostgreSQL, Oracle SQL). For supporting Microsoft Access, it is encouraged to use a language that runs on the .NET Framework Runtime, so the easiest options were Visual Basic and C#.

Király, Balla, & Király, 2022) and (Kádek, Kósa, Orosz, & Tóth, 2017) have already by using only SQL commands, and checking their results is sufficient for the evaluation (and also in the entire

local database management system, PostgreSQL, Oracle SQL, a language that runs on the .NET Framework Basic and C#.

(Király, Balla, & Király, 2022) and (Kádek, Kósa, Orosz, & Tóth, 2017) have concluded that by using only SQL commands, and checking their results is sufficient during black-box testing. We also came to this conclusion but realized that query types that is required on the school leaving examination (and also in the entire National Core Curriculum) can be examined by only supporting a few possible steps: Getting the name (which subtask is solved in it) and SQL-script from a submitted query.

place the content of a table with predefined values.

and contents of a table and later check its correctness.

- Replacing the content of a table with predefined values.
- Storing the structure and contents of a table and later check its correctness.
- Storing the results of a submitted SQL query (with its structure and contents) to later check its correctness.
- Executing a submitted SQL-script.
- Restore the entire database (its schema and the contents of the tables).

Implementation of step *a.* is easy for every database-management system, *b.* can use a simple data frame and convert it to the specific requirements of the given database⁵, *c.* and *d.* can be saved to a simple data frame and easily checked, *e.* should

⁵ It usually can be done by truncating the given table and then inserting multiple rows at once. With Microsoft Access it can be done efficiently by creating an external database

be quite trivial given that a . has been implemented, f .’s implementation depends heavily on the database-management software’s capabilities⁶ but generally speaking it is not difficult.

We will discuss below how DQL, DML, DDL queries can be checked:

When a query written in data query language (SELECT) needs to be checked, steps of b . is used to create a testcase, then d . is used to check the correctness of the query.

When a query written in data manipulation language (INSERT, UPDATE, DELETE) needs to be checked, steps of b . are used to create a testcase, then e . is used to run the submitted DML query, then steps of c . is used to check the correctness of the query. It should be noted that DML queries are not expected to make changes to the database schema, so newer testcases can be prepared using only steps of b . We also added a step of f . in the end.

When a query of data definition language (CREATE, DROP, ALTER, TRUNCATE) needs to be checked, steps of b . is used to create a testcase, the e . is used to run the submitted DDL query, then steps of c . is used to check the correctness of the query, then a step of f . restores the structure of the database so newer testcases can be prepared using only steps of b .

For DQL queries, we reference it as “Queries”, and refer to DML and DDL queries as “Action queries” in some of the screenshots.

Essentially, we have created an abstraction layer for database management systems while still being able to check queries that are required on the school leaving examination in Hungary.

SCREENSHOTS AND VIDEO

The Task Management System (<https://tms-elte.gitlab.io/>) is a learning management system that is used on our faculty to do black-box testing of code submitted by our students. Our solution has been partially integrated (MySQL can be checked) with a local installation of TMS which is shown in (Horcsin, 2023). This software is in progress.

and load a table from there (our task preparator software creates it automatically). With Oracle Database we used the INSERT ALL statement.

⁶ Since restoring the original state of the database is not a security-critical aspect of the system because the database-management software’s process is sandboxed, we have quite a few options. For MariaDB, MySQL, Oracle SQL the schema is dropped, and created again with the preparation script provided with each task. For Microsoft Access a new copy of the original database file is created.

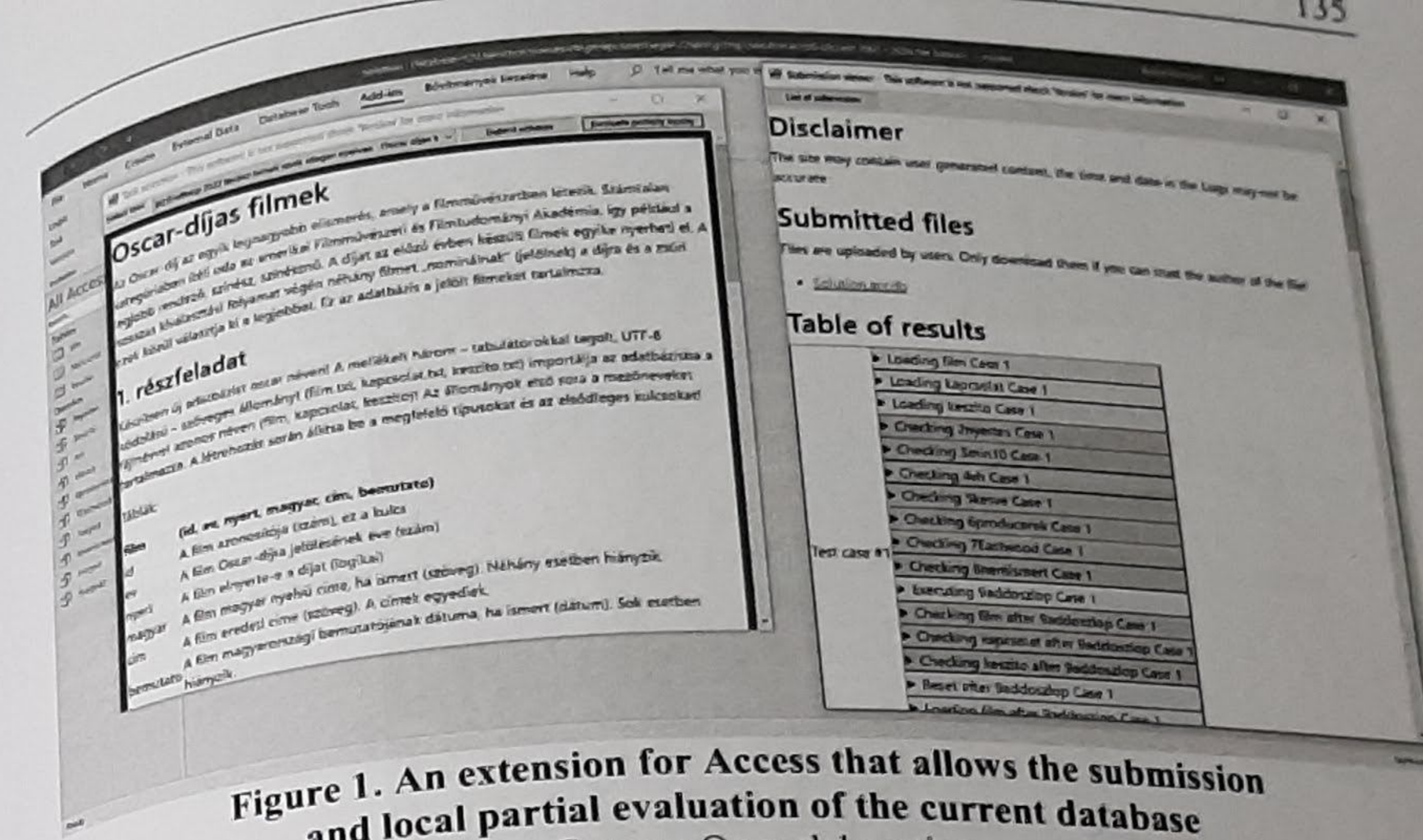


Figure 1. An extension for Access that allows the submission and local partial evaluation of the current database
Source: Own elaboration

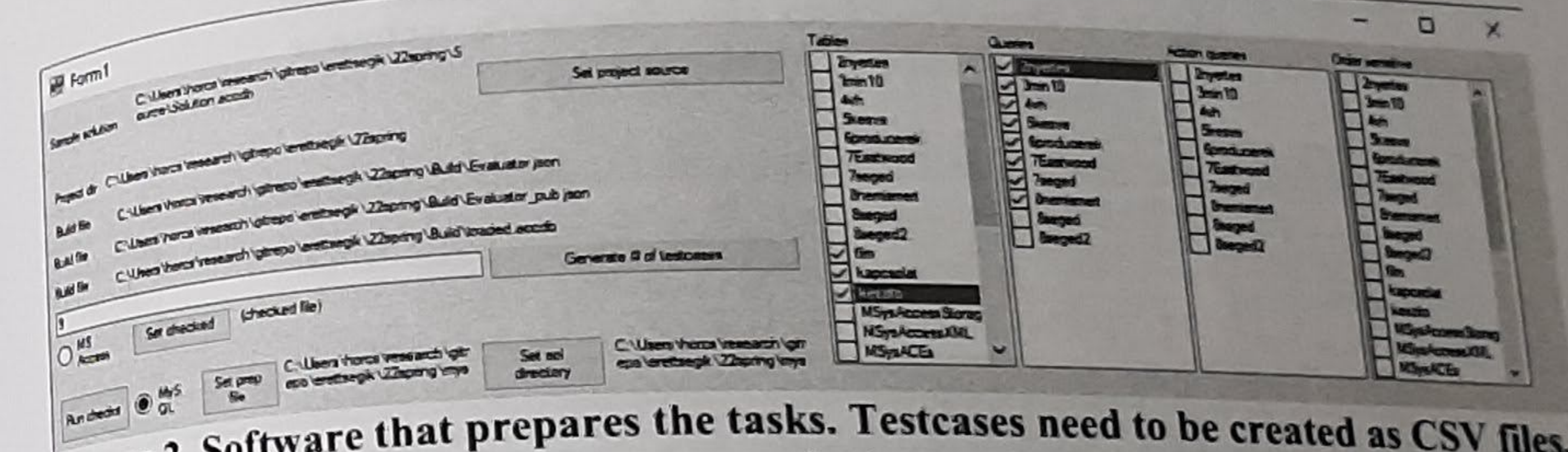


Figure 2. Software that prepares the tasks. Testcases need to be created as CSV files. The number of testcases, a correct solution needs to be given to the software. And the name of tables that need to be truncated and rewritten in each case, the name of checked DQL, DML and DDL queries needs to be selected. And also if the checking should require a specific order of records
Source: Own elaboration

Table of results

Test case #1	▶ Loading film Case 1
	▶ Loading kapcsolat Case 1
	▶ Loading készito Case 1
	▶ Checking 2nyertes Case 1
	▶ Checking 3min10 Case 1
	▶ Checking 4vh Case 1
	▶ Checking 5kesve Case 1
	▶ Checking 6producerek Case 1
	▶ Checking 7Eastwood Case 1
	▶ Checking 8nemismert Case 1
Test case #2	▶ Loading film Case 2
	▶ Loading kapcsolat Case 2
	▶ Loading készito Case 2
	▶ Checking 2nyertes Case 2
	▶ Checking 3min10 Case 2

Figure 3. Overview of the checking on the webpage
Source: Own elaboration

Test case #1 Loading Film Case 1					
id	ev	nyert	magyar	clm	bemutato
5391596	2004	1	NULL	Optimized real-time standardization	1994-8-20
8072445	1998	1	Béke utca kék	Universal national interface	1974-8-26
471400	1986	1	NULL	Universal encompassing synergy	1978-6-18
599375	2003	0	Arany János utca bordó	Centralized demand-driven interface	1995-1-14
4025252	1999	0	Ady endre utca fehér	Profit-focused holistic attitude	2008-6-2
5405282	2004	1	Hídnádasdi utca citromzöld	Object-based well-modulated strategy	2006-5-27
8166496	1986	1	NULL	Assimilated user-facing moderator	1978-5-10
2179119	2008	1	Györgyvégi út bordó	Automated human-resource approach	1974-6-12
2359057	1998	1	NULL	Intuitive user-facing capacity	2002-9-14
4311881	2004	1	NULL	Multi-tiered bifurcated service-desk	1976-12-13
472600	1986	1	Györgykövesdi út kék	Managed global migration	1985-6-25
4982755	1985	0	Oroszgyörgyi utca akvamarin	Progressive next generation success	NULL
386120	2004	0	Arany János utca királykék	Networked multi-state Graphic Interface	NULL

Figure 4. Showing the contents of the testcase

Checking 2nyertes Case 1
Probably used query (2nyertes):
SELECT film.ev, film.clm
FROM film
WHERE (((film.nyert)=True))
ORDER BY film.ev, film.clm DESC;

Answer from the submitted code		Correct solution	
ev	clm	ev	clm
1970	progressive incremental moratorium	2004	Optimized real-time standardization
1973	Reactive encompassing toolset	1998	Universal national interface
1973	Object-based bifurcated portal	1986	Universal encompassing synergy

Figure 5. Showing the submitted code's answer and the correct solution
Source: Own elaboration

Submissions

7602	2023-04-09 00:02:13.611986	EVALUATED	5
7601	2023-04-09 00:02:13.604231	EVALUATED	5
7600	2023-04-09 00:02:13.596582	EVALUATED	5
7599	2023-04-09 00:02:13.588668	EVALUATED	5
7598	2023-04-09 00:01:24.087267	EVALUATED	5
7597	2023-04-08 23:55:54.422554	EVALUATED	5
7596	2023-04-08 23:55:45.555315	EVALUATED	5
7595	2023-04-08 23:54:08.203309	EVALUATED	5
7594	2023-04-08 23:46:55.695995	EVALUATED	5
7593	2023-04-08 23:46:55.687299	EVALUATED	5

Figure 6. List of submissions for a user
Source: Own elaboration

Task: Id: 5 Title: Érettségi 2022 tavasz (emelt szint), idegen nyelven - Oscar-díjas filmek

Select software: MS Access

Select file to upload:

Choose Files No file chosen

Submit solution Upload file

Oscar-díjas filmek

Az Oscar-díj az egyik legnagyobb elismerés, amely a filmművészetben létezik. Számtalan kategóriában ítéli oda az amerikai Filmművészeti és Filmtudományi

Figure 7. Page of a task

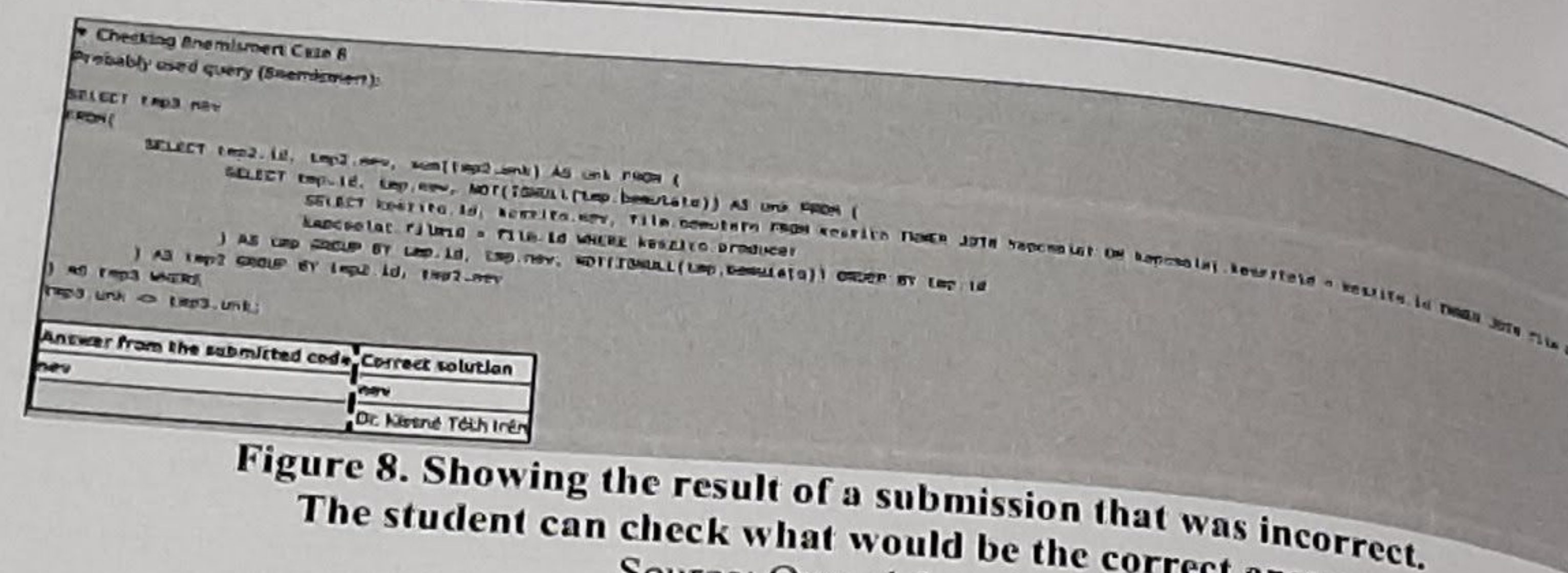


Figure 8. Showing the result of a submission that was incorrect. The student can check what would be the correct answer. Source: Own elaboration

CONCLUSION

Our method makes it easy to create tasks about database management for multiple target audiences (high-school students preparing for the intermediate level school leaving exams, high-school students preparing for the advanced level school leaving exams and university students learning databases).

Running internal unit tests for SQL queries has allegedly been a practice in the industry⁷, and they do use sandboxing and containerization (for example Docker). Using the Linux kernel's capabilities to provide a safe environment for untrusted code to be executed is also common practice around the industry⁸ and also in the space of automatic grading according to the pages of CMS and IOI / isolate, there are also numerous entities providing virtual private servers (VPS providers), which are essentially virtual machines where the tenant can execute arbitrary code.

Compared to (Kádek, Kósa, Orosz, & Tóth, 2017) our method requires only a few well-described functions for each database-management system to be implemented, and the introduction of sandboxing makes it possible to later introduce support for checking queries written in data control language (DCL) or transaction control language (TCL). Sandboxing is a more established way of running untrusted code in the perspective of IT security than running them inside a database transaction. And our method allows the creation of a new task to be less tedious.

Compared to (Király, Balla, & Király, 2022) our method supports multiple flavors of SQL, and the benefits of using sandboxing instead of temporary tables are similar to using sandboxing instead of transactions.

Using our methods allows the creation of a system where the user interface clearly shows to every student how their code made a mistake and which subtask they should improve.

⁷ Discussed here: <https://github.com/oracle/docker-images/issues/1862>.

⁸ For example, AWS' Lambda also uses these capabilities and Firecracker according to [\[https://docs.aws.amazon.com/whitepapers/latest/security-overview-aws-lambda/lambda-isolation-technologies.html\]](https://docs.aws.amazon.com/whitepapers/latest/security-overview-aws-lambda/lambda-isolation-technologies.html).

Plans and possibilities for dissemination

We are planning on using the method described above on our faculty on our databases courses starting on the 2023/2024/2 semester. We are also planning on making the system available for high-school students if it succeeds on our faculty and if we get the necessary funds to deploy the servers. The first open-source release of the software checking databases is planned to be done in the year of 2024 on the website <https://dbtester.elte.hu/>.

We are planning on making it possible to check procedural code as well since our faculty teaches PL/SQL scripts on the "Databases I" course, which would partially hinder the agnosticism (would not be supported on Microsoft Access).

There should be done further research to determine if it is possible to check the performance of a SQL script with great accuracy. We are also planning on assessing the effects of this learning management system by introducing it to some groups in our faculty and comparing their exam scores with the ones in the control group. We are also planning on doing research on white-box testing and accommodate it inside the platform.

BIBLIOGRAPHY

- Dekeyser, S., de Raadt, M., & Lee, T. Y. (n.d.). Computer Assisted Assessment of SQL Query Skills. doi:10.5555/1273730.1273737.
- Fong, C. J., Patall, E. A., Vasquez, A. C., & Stautberg, S. (2018). A Meta-Analysis of Negative Feedback on Intrinsic Motivation. *Educational Psychology Review* volume. Springer US. doi:10.1007/s10648-018-9446-6.
- Horesin, B. (2023, July 29). DB evaluator – TMS integration – demo. Retrieved August 26, 2023, from <https://youtu.be/urkxfBh5PtU>.
- Horváth, G., Horváth, G., & Zsakó, L. (2017). A BÍRÓ és a MESTER – az online értékelés szerepe a programozás oktatásban. *Mérési és értékelési módszerek az oktatásban és a pedagógusképzésben* (ISSN 2416-2957), 89-102. (K. Krisztina, & H. Zoltán, Eds.) ELTE Eötvös Kiadó. Retrieved August 26, 2023, from https://www.eltereader.hu/media/2017/07/Diszciplina_5_READER.pdf.
- Kádek T., Kósa M., Orosz A., & Tóth R., (2017), Új fejlesztési irányok programozási feladatok megoldáskiértékelő rendszereihez. *INFODIDACT 2017* (ISBN: 978-615-80608-1-3). (P. Szlávi, & L. Zsakó, Eds.) Webdidaktika az Oktatásért és az Információs Társadalomért Alapítvány. Retrieved August 26, 2023, from <http://konferenciak.inf.elte.hu/infodidact/InfoDidact17/Manuscripts/KTKMOATR.pdf>.
- Király S., Balla T., & Király R., (2022), Learning SQL by practicing on popular movie databases. *Central-European Journal of New Technologies in Research, Education and Practice*, 4(1), 16-24. doi:10.36427/CEJNTREP.4.1.4465.
- Korom, R. (n.d.). Adatbázis modellezés és SQL feladatok automatikus kiértékelése gráfelméleti alapokon. Retrieved August 26, 2023, from https://www.inf.u-szeged.hu/sites/default/files/koromrichard_tdka2020o.pdf.

- Ribeiro, P., & Guerreiro, P. (2009). Improving the Automatic Evaluation of Problem Solutions in Programming Contests. *Olympiads in Informatics*, 3, 132-143. Retrieved August 26, 2023, from <https://ioinformatics.org/journal/INFOL048.pdf>.
- Shah, B., & Pareek, J. (2022). Automated Evaluation of SQL Queries: Eval_SQL. *Evolution in Computational Intelligence, Proceedings of the 9th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA 2021)*, 89-104. doi:10.1007/978-981-16-6616-2_9.

NOVEL DATABASE MANAGER AGNOSTIC METHOD FOR AUTOMATICALLY CHECKING THE CORRECTNESS OF A SUBMISSION FOR A DATABASE MANAGEMENT TASK

Abstract: Databases play an important role in modern computer systems. From universal database management systems serving web applications to OCHA's Humanitarian Data Exchange providing crucial geographic information for the emergency responders in case of humanitarian crises.

Database management systems are in the National Core Curriculum in Hungary where pupils usually learn the basics of these technologies using Microsoft Access or LibreOffice Base. The curriculum has started to gradually change in 2020 which transition will be completed this year (2023). In information technology – alongside multiple other changes – programming and database management has become more important in the curriculum. Pupils participating in the advanced level matura examination will have to use MariaDB.

According to results from previous years, we have concluded that creating an openly available system that helps pupils to learn database management and to help them switch between office software and professional systems is crucial. It was also important for us to be able to use these advancements in our university.

There have been several successful attempts at creating systems that help students with learning SQL. These systems usually had some written materials and some tasks that would be automatically checked by the system. But all those systems heavily relied on having to only accept solutions in a particular language.

While checking database management tasks, we have concluded that queries which types are similar can usually be checked using the same steps. Therefore, by abstracting the steps of checking a task, we could make checking more uniform, and easier to develop.

This abstraction also helped create tasks for multiple database management systems.

Using our method, to create tasks for 5 different flavors of SQL (Microsoft Access, LibreOffice, MariaDB, PostgreSQL, Oracle SQL), the user only needs to create the content of the tables for each test case, select the names for SELECT

queries, and queries updating the database. The checking is done in a sandboxed environment which is deleted once the checking finishes.

Our research shows a novel database manager agnostic method for automatically checking the correctness of a submission for a database management task.

Keywords: databases, computer assisted learning and assessment, SQL.