МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ В ПРИРОДНИЧИХ НАУКАХ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

DOI: 10.31319/2519-8106.1(48)2023.279990
UDC 004.4

Yatsyshyn Vasyl, Candidate of Technical Sciences, Associate Professor, Associate Professor of the Department of Computer Systems and Networks
Яцишин В.В., кандидат технічних наук, доцент, доцент кафедри комп'ютерних систем та мереж
ORCID: 0000-0002-5517-6359
e-mail: vyatcyshyn@gmail.com

Pastukh Oleh, Doctor of Technical Sciences, Professor, Professor of the Department of Software Engineering
Пастух О.А., доктор технічних наук, професор, професор кафедри програмної інженерії
ORCID: 0000-0002-0080-7053
e-mail: oleg.pastuh@gmail.com

Zharovskyi Ruslan, Candidate of Technical Sciences, senior lecturer of the Department of Computer Systems and Networks
Жаровський Р.О., кандидат технічних наук, старший викладач кафедри комп'ютерних систем та мереж
ORCID: 0000-0002-1159-3417
e-mail: zharikua@gmail.com

Shabliy Nataliya, assistant of the Department of Computer Systems and Networks
Шаблій Н.Р., асистент кафедри комп'ютерних систем та мереж
ORCID: 0000-0002-1125-4757
e-mail: natalinash@gmail.com

Ternopil Ivan Puluj National Technical University, Ternopil
Тернопільський національний технічний університет імені Івана Пулюя, Тернопіль

SOFTWARE TOOL FOR PRODUCTIVITY METRICS MEASURE OF RELATIONAL DATABASE MANAGEMENT SYSTEM

ПРОГРАМНИЙ ЗАСІБ ВИМІРЮВАННЯ МЕТРИК ПРОДУКТИВНОСТІ РЕЛЯЦІЙНИХ СИСТЕМ КЕРУВАННЯ БАЗАМИ ДАНИХ

The paper presents the main results of automating the process of measuring performance metrics of relational database management systems. The implemented software tool is based on the principles of technology for evaluating the performance of software systems, which is based on the formalized quality model of the ISO/IEC 25010 standard and the processes of planning and implementing the evaluation procedure. The article offers a formal description of performance *characteristics at the level of sub-characteristics, attributes and metrics. The architecture and scheme of the database of the software tool for measuring performance indicators were designed and implemented as a software tool.*

***Keywords**: software tool, measure, metric, productivity, database management system.*

Сучасний розвиток IT галузі характеризується розробкою і впровадженням нових та дієвих інструментів, що орієнтовані на підвищення рівня автоматизації виконуваних задач і процесів життєвого циклу як елементарних програмних систем, так і складних комп'ютерних комплексів. До складу більшості комп'ютерних систем входять бази даних, а однією з ключових характеристик якості на сьогодні є продуктивність, тому актуальності набуває розробка програмного засобу вимірювання метрик атрибутів продуктивності реляційних систем керування базами даних. Такий засіб забезпечить можливість автоматизованого вимірювання показників продуктивності при експертному оцінюванні реляційних СКБД та знизить поріг входу і складність виконання операцій для експертів з оцінювання якості.

У роботі, на основі технології оцінювання якості програмних систем та моделі якості стандарту ISO/IEC 25010, запропоновано структуру концептуальної моделі оцінювання реляційних систем керування базами даних та реалізовано програмний засіб вимірювання метрик продуктивності реляційних СКБД. Засіб автоматизації процесу оцінювання продуктивності СКБД володіє такими функціями як: внесення та збереження вихідних даних (підключень, інформацію про сервери СКБД, атрибути, метрики, готовий до використання код чи його шаблони); налаштування тесту, збереження налаштувань, завантаження налаштувань; проведення перевірки сумісності налаштувань та обраної СКБД;встановлення з'єднання та проведення тестування;вимірювання необхідних показників, збереження результатів та їх візуалізація. На основі сформованих вимог до систем оцінювання продуктивності СКБД, базуючись на концептуальній моделі продуктивності СКБД та інформації про визначені атрибути, їх метрики і функції оцінювання, розроблено трьохрівневу архітектуру засобу автоматизації. Спроектована архітектура дозволила забезпечити реалізацію простого і зрозумілого засобу підтримки експертного оцінювання продуктивності СКБД.

Проведено експериментальні дослідження, які передбачали вимірювання метрик продуктивності для двох реляційних СКБД: Oracle 11g та MS SQL Server. Результати експерименту підтвердили доцільність використання розробленого програмного засобу, оскільки забезпечують зниження витрат часу і порогу входу для експертів при проведенні процедур оцінювання продуктивності СКБД.

***Ключові слова**: програмний засіб, вимірювання, метрика, продуктивність, система керування базами даних.*

Problem's Formulation

The modern development of the IT industry is characterized by the development and implementation of new and effective tools aimed at increasing the level of automation of performed tasks and life cycle processes of both elementary software systems and complex computer complexes. To reduce the complexity and increase the efficiency of the implementation of computer systems, the approach of reuse of components is quite often used [1]. However, this approach requires procedures for assessing the quality of finished components for their possible use in a specific project. Since most computer systems include databases, and one of the key quality characteristics today is performance, the development of a software tool for measuring performance attribute metrics of relational database management systems is gaining relevance. Such a tool will provide the possibility of automated measurement of performance indicators during expert evaluation of relational DBMS and will reduce the entrance threshold and the complexity of operations for quality evaluation experts.

Analysis of recent research and publications

Considering the tendency of increasing complexity of computer systems implementation, it is advisable to use CASE-tools [1—3] in accordance with their purpose and the stage of project implementation. Most tools for automating life cycle processes are "boxed" enterprise solutions [4—6] and quite often are oriented to use in predetermined software development methodologies and life cycle models.

The application and adaptation of such tools when evaluating the performance of database management systems is a rather complex and non-trivial task that involves the involvement of additional software testing specialists. To improve the efficiency of the DBMS performance evaluation process, it is advisable to develop a software tool for measuring relational DBMS performance indicators with a simple user interface that will provide a low entry threshold for software quality evaluation experts.

Formulation of the study purpose

The purpose of the research is to develop a software tool that provides automation of the process of measuring performance attribute metrics of relational database management systems, taking into account the features of quality assessment technology based on the recommendations of the ISO/IEC 25010 standard and ensuring ease of use by quality assessment experts.

Presenting main material

Relational databases today occupy about 80 % of the market among all types of storage, and the selection of the optimal DBMS, from the point of view of performance in the design of computer systems, should be based on a formalized attribute evaluation procedure.

In [7], a formalized technology for evaluating the quality of software systems that belong to the class of web-oriented systems is proposed. When evaluating the performance of relational DBMS, it is suggested to use just such an approach and implement a software tool for automated measurement of performance attribute metrics. The basis of the performance evaluation technology is an expert approach and a quality model, the structure of which is presented in fig. 1.

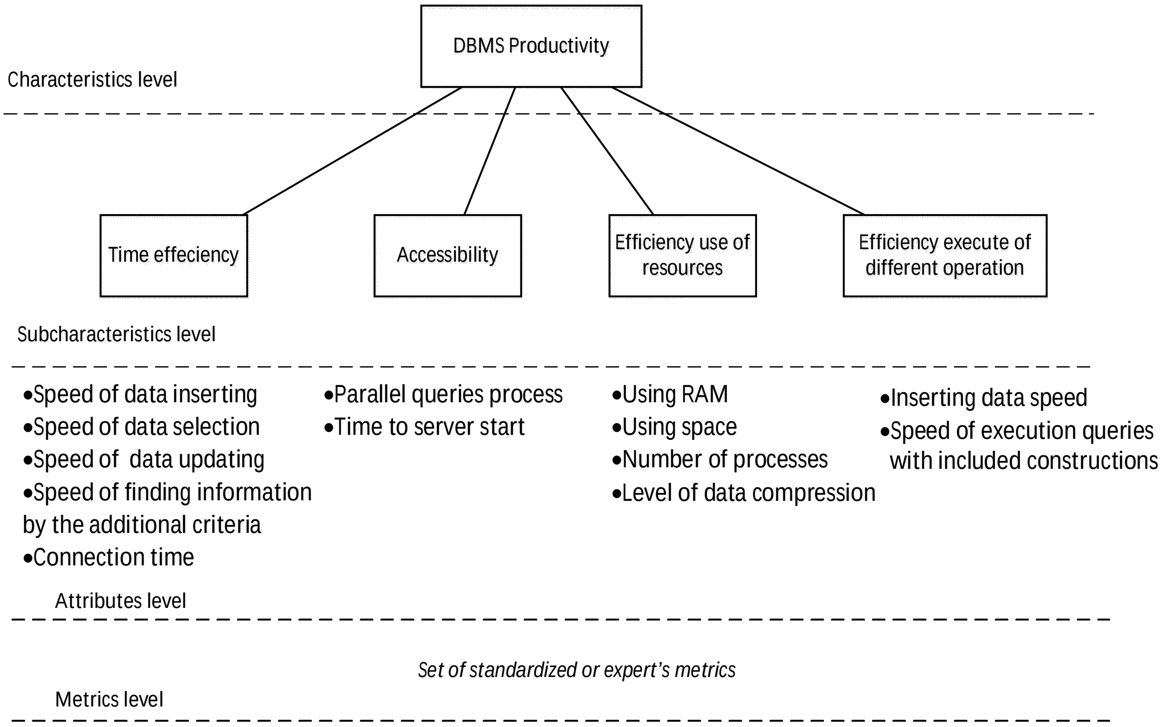


Fig. 1. The structure of the relational DBMS performance evaluation model

An example of a non-formalized but structured description of the performance characteristic attribute is given in the tabl. 1.

Table 1. Description of the «Server startup time» attributes

Characteristic	Productivity
Sub characteristic	Accessibility
The name of the attribute	Server startup time

Continue of the table 1

Attribute definition	The server start-up time is the time from the receipt of the command to start the server to the complete loading of all its components for full performance of the assigned tasks.
Purpose/Motivation	Server startup can be external or internal. External server startup is performed by the user or another application program, and can be local or global, depending on the location of the application program. The internal start of the server is carried out by setting the schedule of its inclusion and is performed by the server itself. The server startup time, regardless of the startup type, should be the same.
Measurement scale	Absolute (c)
Determination procedure, protocol, X	The value of the attribute is determined by the time from submitting the command to start the server to the complete start of all its components.
Type of data collection and counting	Manual, with the help of devices, automated
Interpretation of the value of the assessment	The less, the better

A formalized description of the model used in relational DBMS performance evaluation is shown in Fig. 2.

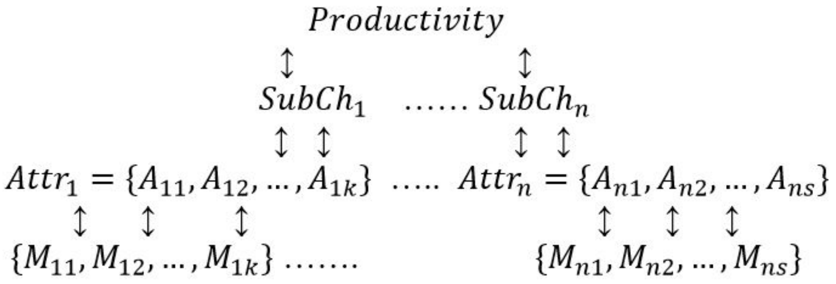

Productivity
SubCh1 …… SubChn
Attr1 = {A11, A12, …, A1k} ….. Attrn = {An1, An2, …, Ans}
{M11, M12, …, M1k} ……. {Mn1, Mn2, …, Mns}

Fig. 2. The formalized structure of the quality model when evaluating DBMS performance

The top of the hierarchy in fig. 2 is $Productivity$ — a comprehensive (integral) performance characteristic of relational DBMS, at the lower level $\{SubCh_i\}$ — a set of sub-characteristics, $\{\ Attr_{im}\}$ — a set of sub-characteristic attributes, which are selected taking into account the specifics of the subject area, M_{im} — appropriate metrics that can be selected from a standardized list or determined by an expert.

Formally, the performance characteristic can be presented in the form of a tuple (1)

$$Productivity = \{SubCh_i, Attr_{im}, Constr_{im}, M_{im}\}. \qquad (1)$$

To evaluate the performance of relational DBMS, a set of attributes and corresponding metrics and elementary functions are defined, which are presented in the format tabl. 2, when planning the evaluation process. The performance evaluation process is performed according to the procedures given in [7], and the determination of the quantitative values of the metrics is proposed to be calculated using a software tool for measuring the arguments that describe the metric.

An important stage of design, which serves as a certain foundation for further development, is the process of mapping the conceptual model (1) to the specified components and objects of the system. Algorithms of functioning of modern software systems are, as a rule, complex and non-linear, which challenges developers and engineers to build the architecture as efficiently as possible based on the development of certain static interfaces of interaction between its components.

Table 2. Format for describing attributes and performance metrics

Attribute	Metrics	Elementary metric evaluation function
The speed of execution of queries with nested constructions	Speed of execution of complex requests $X = \frac{N}{t_{total}}$, where N — amount of data obtained as a result of writing, t_{total} — total execution time of a query with nested structures.	$g(X) = \begin{cases} 1, if X \to X_{max} \\ \frac{X}{X_{max}}, 0 \le X \le X_{max}, \text{де}\, X_{max} \text{ —} \\ 0, in\ some\ cases \end{cases}$ the largest maximum value for an attribute given by an expert for a specific subject area

The components of the system implement separate elements of the functionality of the software product, combining by providing interaction interfaces, data transmission flows and the use of common data sources, based on the same principles of behavior they form a single complex. A tool for automating the DBMS performance evaluation process should include the following features:

- entering and saving source data (connections, information about DBMS servers, attributes, metrics, ready-to-use code or its templates);
- test settings, saving settings, loading settings;
- checking the compatibility of settings and the selected DBMS;
- connection establishment and testing;
- measuring the necessary indicators, saving the results and their visualization.

Based on the formed requirements for DBMS performance evaluation systems, based on the conceptual model of DBMS performance and information about the defined attributes, their metrics and evaluation functions, the following three-level architecture of the automation tool was developed (Fig. 3).

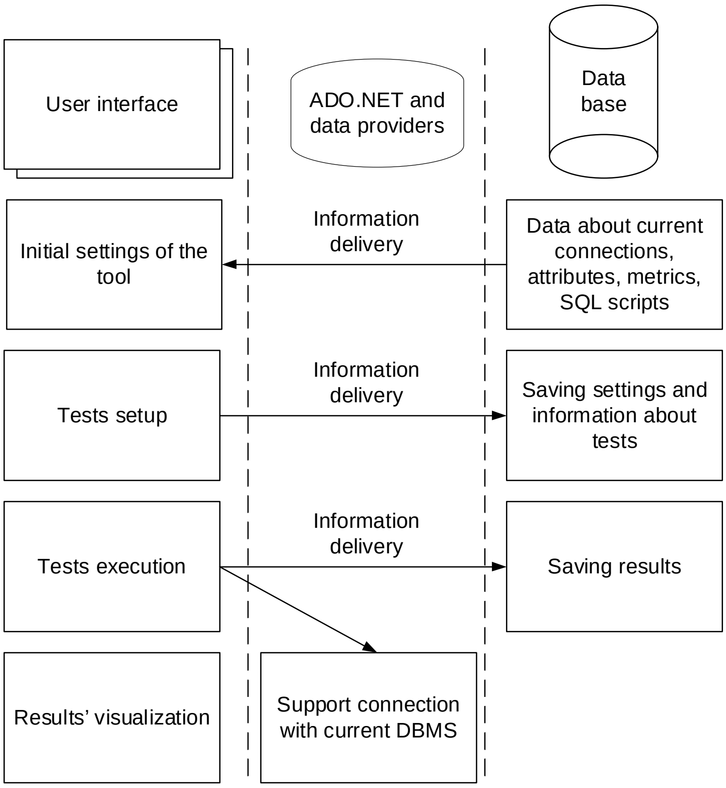

User interface
ADO.NET and data providers
Data base
Initial settings of the tool
Information delivery
Data about current connections, attributes, metrics, SQL scripts
Tests setup
Information delivery
Saving settings and information about tests
Tests execution
Information delivery
Saving results
Results' visualization
Support connection with current DBMS

Fig. 3. Architecture model of the DBMS evaluation process automation tool

The user interface layer provides the graphical interface and associated code to perform the following functions:

- initial setting of the tool is the selection of available DBMS, display of system information, verification of connections to selected database servers, visualization of additional information;
- setting up the tests — consists in choosing the attributes to be tested, the method of their testing, entering additional data (the date of the assessment, the details of the expert, additional software code);
- conducting tests — the process of evaluating the selected DBMS according to the methods specified by the user, measuring indicators, and processing them;
- visualization of results — display of evaluation results in a form acceptable to the expert.

The ADO.NET and data providers layer provides data transfer between other layers of the architecture and provides a mechanism for launching and executing test code in target DBMS.

The database layer provides the storage of raw data for tests such as:

- available connections;
- information about DBMS;
- performance attributes;
- metrics and information about their interpretation;
- test program code and information about its compatibility with various DBMS.

Also, its functions include saving and loading settings, information about tests, saving test results.

The division of the architecture into three levels allows to guarantee the transparency of their interaction, easy replacement of software structures, and ease of modernization of the tool on demand.

Relational databases are most often used to store data used by software, to retrieve, process, and update them. Relational databases are built on the basis of domain objects and relationships between them. Such an analysis is performed on the basis of an assessment of the specifics of the tasks and the purpose of the development.

After conducting an analysis of the subject area, its entities and connections between them, a database scheme was built, the ER-diagram of which is shown in fig. 4.

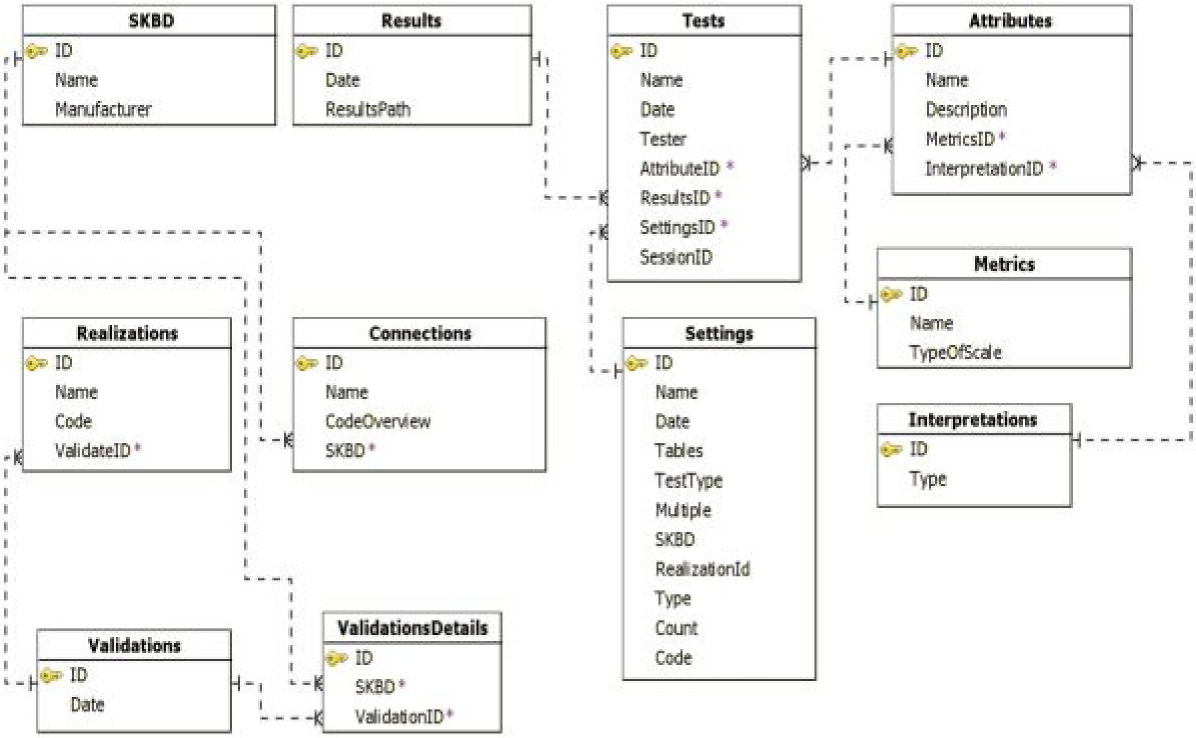


Fig. 4. ER diagram of the database

When designing the database, the requirements for ensuring its integrity and normalization are taken into account. Data provider .NET OLE DB uses integrated OLE DB COM access to connect to databases. It supports local or remote, single or distributed connections to the database management system.

The graphical interface plays an important role in ensuring the interaction between the user and the software. The software interface should provide maximum simplicity and comprehensibility with complete functionality. These requirements are satisfied due to the convenient location of control elements, the presence of clear and simple labels on buttons and other components, structured presentation of large data sets (for example, in the form of tables).

Taking into account the specifics of using the tool for automating the process of evaluating the performance of relational DBMS, a simple and functional graphical user interface has been implemented, the start window of which is shown in Fig. 5.

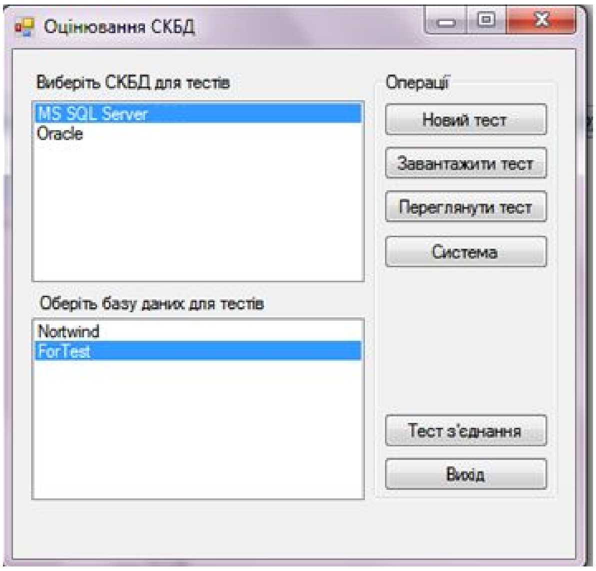


Fig. 5. View of the start window of the software tool

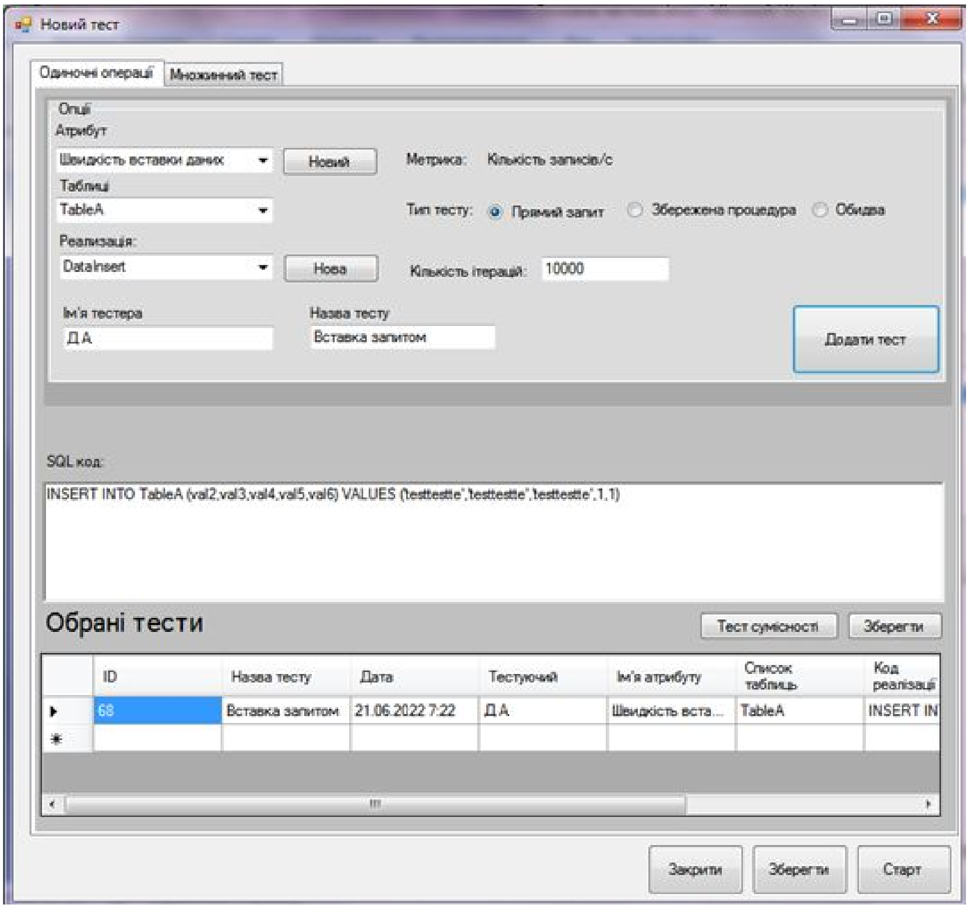
In the left part of the window (Fig. 5) above and below there are two lists with available connections. When you select a DBMS in the top list, the bottom list will display the databases available for that DBMS. There are two groups of buttons on the right. The upper group contains buttons for starting the test, downloading parameters, viewing results, and viewing system information. The lower group of buttons allows you to test the connection with the base or to terminate the program. When you click the "New test" button of the main window, the DBMS performance evaluation setting window opens (Fig. 6). This is the main window of the program, contains many controls, but is convenient due to the structuring.

In the upper part of the form there are drop-down lists and text fields for entering test parameters, namely:

- performance attributes and metrics;
- tables;
- type of test;
- SQL query implementation;
- the name of the tester (expert) and the name of the request.

Below is a field for entering or editing an SQL statement. Below it is a DataGridView element that displays the test data of the tests of the current session.

After pressing the "Start" button, the test data will be transferred to the DBMS performance evaluation modules, which will process queries and perform database manipulations. After that, two windows with data results are opened — in the form of a graph and in the form of a table (Fig. 7 and Fig. 8). When you click the "Save to file" button, a dialog for saving the received data to a text file appears on the form with a tabular representation of the results.



Новий тест

Атрибут: Швидкість вставки даних

Метрика: Кількість записів/с

Таблиці: TableA

Тип тесту: ● Прямий запит ○ Збережена процедура ○ Обидва

Реалізація: DataInsert

Кількість ітерацій: 10000

Ім'я тестера: Д А

Назва тесту: Вставка запитом

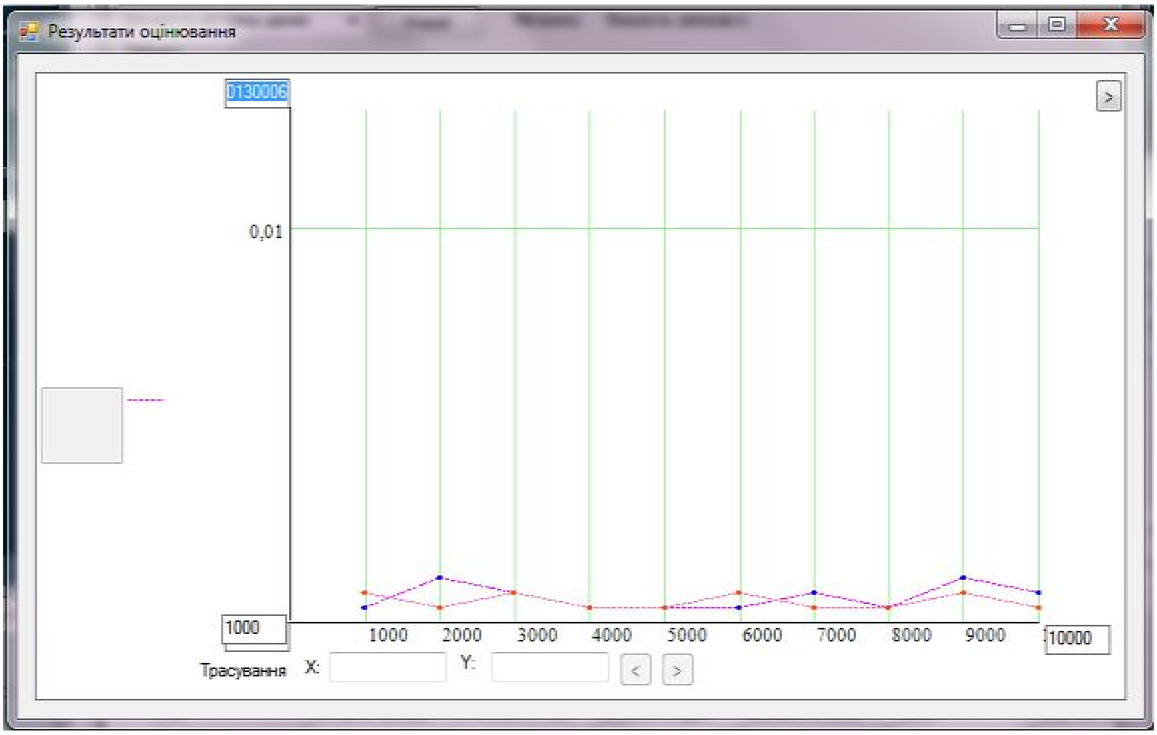
SQL код:

```
INSERT INTO TableA (val2,val3,val4,val5,val6) VALUES ('testtestte','testtestte','testtestte',1,1)
```

Обрані тести

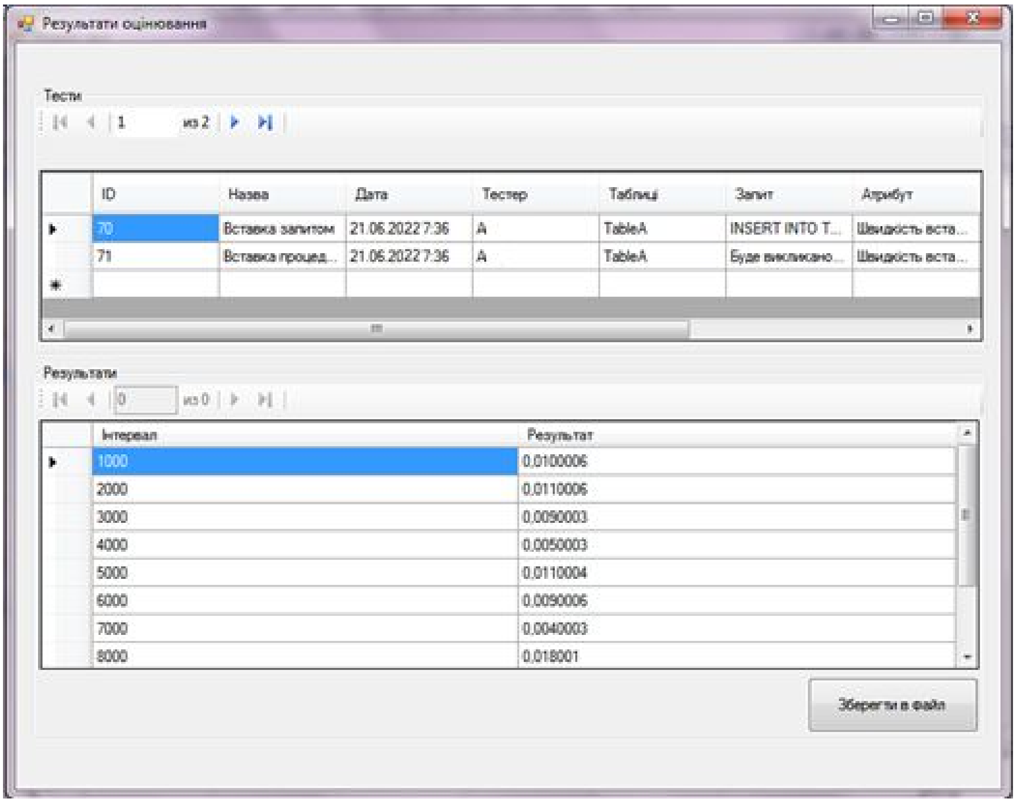
	ID	Назва тесту	Дата	Тестуючий	Ім'я атрибуту	Список таблиць	Код реалізації
▸	68	Вставка запитом	21.06.2022 7:22	Д А	Швидкість вста...	TableA	INSERT IN
*							

Fig. 6. The form for configuring parameters of DBMS performance evaluation



Результати оцінювання

Fig. 7. Visualization of the evaluation results in the form of a graph



Результати оцінювання

Тести

ID	Назва	Дата	Тестер	Таблиці	Запит	Атрибут
70	Вставка запитом	21.06.2022 7:36	A	TableA	INSERT INTO T...	Швидкість вста...
71	Вставка процед...	21.06.2022 7:36	A	TableA	Буде викликано...	Швидкість вста...

Результати

Інтервал	Результат
1000	0,0100006
2000	0,0110006
3000	0,0090003
4000	0,0050003
5000	0,0110004
6000	0,0090006
7000	0,0040003
8000	0,018001

Fig. 8. Visualization of the evaluation results in the form of a table

If you click "Load test", the initial form displays a window with a list of tests sorted by session identifiers and date. When selecting a table entry and pressing the "Load" button, the "New test" window opens with the installed options and added tests.

After analyzing the main advantages and disadvantages of relational DBMS, Oracle 11g and MS SQL Server were chosen as the target environments for the evaluation. This choice was made on the basis of the prevalence of DBMS data, good support by development tools, their support of the main concepts of RDBMS, high indicators of TRS tests [8].

The tests were performed on relational tables with the same field structure: int/number(10), varchar(10)/varchar2(10), varchar(max)/varchar2(max), char(10),bit/number(1),float. The attributes measured in the experiment were "Writing speed" and "Data sampling speed". The number of measurement iterations is 10,000 when using direct and stored procedure calls. Received data presented in tabl. 3 and tabl. 4.

Table 3. Results of data recording based on direct calls

Range of iterations	Oracle 11, recording time, ms	MS SQL recording time, ms
1000	0,003	0,009
2000	0,02	0,007
3000	0,02	0,008
4000	0,01	0,009
5000	0,004	0,007
6000	0,005	0,012
7000	0,02	0,01
8000	0,006	0,014
9000	0,004	0,007
10000	0,009	0,008
Average value	0,0101	0,0091

Table 4. Results of data sampling based on direct calls

Range of iterations	Oracle 11g, sampling time, ms	MS SQL Server, sampling time, ms
1000	0,007	0,004
2000	0,007	0,006
3000	0,005	0,005
4000	0,004	0,006
5000	0,007	0,005
6000	0,005	0,008
7000	0,008	0,004
8000	0,007	0,006
9000	0,009001	0,004
10000	0,007	0,008
Average value	0,0065	0,0056

As a result of the conducted research, a software tool for measuring performance attribute metrics, based on the technology for evaluating the quality of software systems proposed in [7], was implemented in practice. This made it possible to automate the procedures for expert evaluation of the performance of relational database management systems when selecting them as components of more complex computer systems based on the approach of component reuse. In addition, as a perspective for the development of the approach to the evaluation of the quality of software systems in the context of the automation of relevant processes, it is planned to integrate the developed tool for measuring metrics with the CASE-tool for software quality management [9].

Conclusions

The main scientific and practical results of this article are as follows:

1. On the basis of technology [7] and the quality model of the ISO/IEC 25010 standard, the structure of the conceptual model of software evaluation belonging to the class of database management systems was built, which made it possible to determine the main components of the software tool for measuring performance attribute metrics of relational DBMS.
2. The three-level architecture and database scheme of the software tool for measuring the values of metrics of the relevant attributes were designed using WinForms, ADO.NET and MS SQL Server technology, which made it possible to practically implement a simple and understandable tool for supporting expert evaluation of DBMS performance.
3. Experimental studies were conducted to determine the performance metrics of two relational DBMS: Oracle 11g and MS SQL Server, which confirmed the feasibility of developing and using this tool, as it reduces time costs and the entrance threshold for experts when conducting evaluation procedures.

Referneces

[1] Ayad, I., Jameel A., Aysh, Al (2015). The Definition of Intelligent Computer Aided Software Engineering (I-CASE) Tools. Journal of Information Engineering and Applications Vol.5, No.1.

[2] Poulding, S. M., Clark, J. A. (2010). Efficient Software Verification: Statistical Testing using Automated.Search. IEEE Transactions on Software Engineerig.

[3] Thomas, S., Adams, B., Hassan, A., Blostein, D. (2014). Studying software evolution using topic models. Sci. Comput. Program., vol. 80, Part B.

[4] IBM Rational DOORS”. http://www-01.ibm.com/software/ awdtools/doors/

[5] IBM Rational RequisitePro. http://www -01.ibm.com/ software/awdtools/reqpro/

[6] IBM Rational System Architect Overview. https://www.peerspot.com/products/ibm-rational-system-architect-reviews.

[7] Yatsyshyn, V.V. (2009). Tekhnolohiya otsinyuvannya yakosti WEB-zastosuvan' [Technology of web application quality evaluation]. Visnyk Ternopil's'koho derzhavnoho tekhnichnoho universytetu.

[8] Kharchenko, A., Galay, I., Yatcyshyn, V. (2011) The method of quality management software. 2011 Proceedings of 7th International Conference on Perspective Technologies and Methods in MEMS Design, MEMSTECH 2011 . Polyana.

[9] Yatsyshyn, V.V., Kharchenko, O.H. (2010) CASE-tekhnolohiya rozroblennya vymoh do prohramnoho zabezpechennya ta otsinyuvannya yoho yakosti [CASE-technology of software requirements development and its quality evaluation]. Naukovyy visnyk NLTU Ukrayiny. 20(2).

Список використаної літератури

1. Ayad I., Jameel A., Aysh Al. The Definition of Intelligent Computer Aided Software Engineering (I-CASE) Tools. Journal of Information Engineering and Applications. 2015. Vol.5, No 1. P. 47–56.
2. Poulding S.M., Clark J.A. Efficient Software Verification: Statistical Testing using Automated Search. IEEE Transactions on Software Engineerig. 2010. Vol. 36 (6). P. 763–777.
3. Thomas S., Adams B., Hassan A., Blostein D. Studying software evolution using topic models. Sci. Comput. Program. 2014. Vol. 80, part B. P. 457–479.
4. IBM Rational DOORS". URL: http://www-01.ibm.com/software/ awdtools/doors/ (дата звернення: 15.02.2023 р.).
5. IBM Rational RequisitePro. URL: http://www -01.ibm.com/ software/awdtools/reqpro/ (дата звернення: 15.02.2023 р.).
6. IBM Rational System Architect Overview. URL: https://www.peerspot.com/products/ibm-rational-system-architect-reviews. (дата звернення: 15.02.2023 р.).
7. Яцишин В.В. Технологія оцінювання якості WEB-застосувань. Вісник Тернопільського державного технічного університету. 2009. Том 14. С.132–140.
8. Kharchenko A., Galay I., Yatcyshyn V. The method of quality management software. Proceedings of 7th International Conference on Perspective Technologies and Methods in MEMS Design, MEMSTECH 2011. Polyana, 2011. P. 82–84.
9. Яцишин В.В., Харченко О.Г. CASE-технологія розроблення вимог до програмного забезпечення та оцінювання його якості. Науковий вісник НЛТУ України. 2010. Том 20 (2). С. 277–285.

Надійшла до редколегії 06.03.2023