UDC 004.65:004.03

СТРАТЕГІЧНІ АСПЕКТИ ВИБОРУ СИСТЕМИ БАЗ ДАНИХ: POSTGRESQL ПРОТИ MICROSOFT SQL SERVER

STRATEGIC CONSIDERATIONS FOR DATABASE SYSTEMS: POSTGRESQL VS. MICROSOFT SQL SERVER

Попов Р. О.[1], Герасимов В. В.[2]
Popov R. O.[1], Gerasimov V. V.[2]
[1,2] Oles Honchar Dnipro National University (Dnipro, Ukraine)
ORCID: [1] https://orcid.org/0009-0003-6982-2993, [2] https://orcid.org/0000-0002-1366-715X
E-mail: [1] popov_r@365.dnu.edu.ua, [2] herasymov_v@365.dnu.edu.ua

DOI: 10.15673/atbp.v17i3.3156

Abstract*. In this paper, we conducted a comprehensive study of two leading relational database management systems: Microsoft SQL Server and PostgreSQL, representing commercial and open-source software development models respectively. We developed several comparison criteria ranging from toolset availability and platform support to syntax and semantic differences between the systems. We performed a systematic analysis of architectural differences, language constructs, and functional capabilities of both systems based on current versions MS SQL Server 2022 and PostgreSQL 17.5.*

The analysis revealed differences in operational capabilities and design philosophies between the two systems, particularly in their handling of Unicode text processing, binary data storage, and cross-database operations. We identified critical divergences in approaches to namespace organization, cross-database query processing, data type implementation, and indexing mechanisms. PostgreSQL's unified approach to data types and its support for advanced features like native JSON operators and array handling contrast with MS SQL Server's more specialized type system and its distinction between clustered and non-clustered indexing structures. We found that PostgreSQL demonstrates superior platform independence and extensibility due to its open architecture, while MS SQL Server provides deeper integration with the Microsoft ecosystem and offers more differentiated index types.

The study also highlighted the educational implications of these differences, particularly in the context of Ukrainian academic institutions where Microsoft products dominate, potentially limiting students' exposure to industry-standard open-source alternatives that offer comparable functionality at reduced institutional costs. We demonstrated that our methodology can be applied to future RDBMS research and can help Ukrainian universities integrate more open-source software into their database curricula.

Анотація*. У цій роботі ми провели всебічне дослідження двох провідних систем управління реляційними базами даних: Microsoft SQL Server та PostgreSQL, які представляють відповідно комерційні та відкриті моделі розробки програмного забезпечення. Ми розробили декілька критеріїв порівняння, починаючи від доступності інструментарію та підтримки платформ до синтаксичних і семантичних відмінностей між системами. Ми виконали систематичний аналіз архітектурних відмінностей, мовних конструкцій та функціональних можливостей обох систем на основі поточних версій MS SQL Server 2022 та PostgreSQL 17.5.*

Аналіз виявив відмінності в операційних можливостях та філософії проектування між двома системами, особливо в їх обробці Unicode-тексту, зберіганні бінарних даних та міжбазових операціях. Ми виявили критичні розбіжності в підходах до організації простору імен, обробки міжбазових запитів, реалізації типів даних та механізмів індексування. Уніфікований підхід PostgreSQL до типів даних та його підтримка передових функцій, таких як нативні JSON-оператори та обробка масивів, контрастують зі спеціалізованою системою типів MS SQL Server та його розрізненням між кластеризованими та некластеризованими структурами індексування. Ми встановили, що PostgreSQL демонструє вищу платформну незалежність та розширюваність завдяки своїй відкритій архітектурі, тоді як MS SQL Server забезпечує глибшу інтеграцію з екосистемою Microsoft та пропонує більш диференційовані типи індексів.

Дослідження також висвітлило освітні наслідки цих відмінностей, особливо в контексті українських академічних установ, де домінують продукти Microsoft, що потенційно обмежує знайомство студентів із

промисловими стандартами відкритих альтернатив, які пропонують порівнянну функціональність за зниженими інституційними витратами. Ми продемонстрували, що наша методологія може бути застосована до майбутніх досліджень РСРБД і може допомогти українським університетам інтегрувати більше відкритого програмного забезпечення до своїх навчальних програм з баз даних.
***Keywords**: database management systems, PostgreSQL, Microsoft SQL Server, RDBMS comparison, open-source databases, commercial databases, SQL compliance, database architecture, comparative analysis.*

***Ключові слова**: системи управління базами даних, PostgreSQL, Microsoft SQL Server, порівняння РСКБД, бази даних з відкритим кодом, комерційні бази даних, стандарт SQL, архітектура баз даних, порівняльний аналіз.*

INTRODUCTION. Microsoft SQL Server (MS SQL) and PostgreSQL are two of the most popular relational database management systems (DBMS). Both of them have big advantages to use: one of them is commercial and backed up by a well-known company, and other is free (and distributed without cost) and part of open source software. As any modern RDBMS, they also support Structured Query Language (SQL) for access.

The **goal** of our paper is to compare MS SQL and PostgreSQL in practice. The **object** of our paper are relational database management systems. The **subject** of our paper is the comparative analysis of two popular RDBMS: MS SQL and PostgreSQL. We have deliberately chosen these systems for two reasons:

1. MS SQL is one of top studied RDBMS in Ukrainian universities.
2. Microsoft products have a strong presence in Ukrainian universities, which often leads to underexposure to free and open-source alternatives.

The **scientifical significance** of our work lies in designing a comparison methodology that could be applied to other RDBMS (and potentially part of it could be applied to non-relational databases). Our work posesses also an *educational significance*, as methodologists can esitmate whether it's relevant or easy to integrate more free software like PostgreSQL in their educational programs.

The **practical significance** of our work lies in analysis of the database management systems (DBMS). Companies or individuals can assess the capabilites of each systems and choose best that meets their needs.

The **tasks** of the paper are:

- Study the history and main features of MS SQL Server and PostgreSQL.
- Design the methodology for comparing two RDBMS.
- Conduct the comparisons and make conclusions.

1. ITERATURE OVERVIEW

Our literature review will begin with the MS SQL Server, as it was released earlier that PostgreSQL (MS SQL was released in 1989 [1], PostgreSQL — in 1995 [2]). Microsoft SQL Servier is a relational database management system produced and maintained by Microsoft. As many Microsoft products, initially MS SQL was based off another product from Sybase company — Sybase SQL Server 3.0 for UNIX and VMS. Since 1989 it have been continuosly growing and expanded with features, outliving a full rewrite [1].

MS SQL comes in several editions, which depends on the year of the product. For example, MS SQL 2022 comes in these editions: Enterprise (premium offering), Standard (basic data management), Web (external hosting), Developer (for extending), Express (entry-level, free) [3].

MS SQL Server is heavily integrated with another Microsoft product — Visual Studio. There are special classes and components for displaying data from MS SQL database that allows for rapid application development [4].

Another interesting feature of MS SQL is the possibility to analyze the data: Business Intelligence, Data Mining, and Online Analytical Processing (OLAP). MS SQL has various algorithms for building rule-based, statistical, and neural models that could classfiy, clusterize and predict data. Even though most of these algorithms were researched and described much earlier than the release date of this RDBMS, still they are quite integrated to the user data supplied to database and allows for easier analysis [5].

Speaking about other RDBMS, PostgreSQL is a powerful open-source relational database management system with roots in academic research. Its development began at the University of California, Berkeley, where it evolved from the Ingres project led by Michael Stonebraker. Originally designed as an object-relational system under the name Postgres, it was later enhanced with SQL support and renamed Postgres95. In 1996, the project transitioned into a global open-source effort, adopting the name PostgreSQL to reflect its new direction and standards compliance [6].

Since then, PostgreSQL has been actively developed by a distributed team of volunteers and contributors. The project gained traction thanks to its open-source model, strong SQL compliance, and extensible architecture. Over time, it matured into a feature-rich database system used in academic, commercial, and government applications. Its emphasis on code quality, community-driven development, and support for complex data types has made it a trusted foundation for data-intensive systems across the world.

Unfortunately, there is currently limited academic research in Ukraine specifically focused on PostgreSQL. Most references to PostgreSQL are found in student-led projects or academic theses where it serves as the backend for developing domain-specific information systems [7], [8]. Its popularity in such contexts stems from its open-source nature, free licensing, and powerful capabilities, making it an attractive choice for educational and practical implementations. However, these works mainly focus on practical applications rather than fundamental research on the database system itself.

One notable exception is the research by Y. R. Hlynka and R. B. Vovk, which investigates PostgreSQL's transaction mechanisms based on the ACID principles (atomicity, consistency, isolation, durability). The study analyzes transaction processing steps—such as BEGIN, COMMIT, and ROLLBACK—and highlights the role of multiversion concurrency control (MVCC) in ensuring data integrity and stability during concurrent operations and system failures [9].

Although there is limited research specifically focused on PostgreSQL in Ukraine, some studies compare different relational database management systems (RDBMS) to evaluate their suitability for various information systems.

One notable example is the paper by Skrypka and Sharov titled "Comparison of relational databases for use in information systems". This study highlights the importance of relational databases like Oracle Database, MySQL, and Microsoft Access for storing and managing data. They find Access best for local systems due to ease of use, MySQL ideal for web applications because it's open-source and fast, and Oracle suited for large, complex enterprise systems [10].

Another general study by T. S. Nikitina and O. I. Morozova presents a tool for benchmarking database systems under various workloads. Their research focuses on both SQL and NoSQL databases, allowing for flexible testing of operations like insert, delete, update, and select. While not focused on PostgreSQL specifically, it offers a broad comparison of DBMS performance across different scenarios [11].

In conclusion, while both MS SQL Server and PostgreSQL are widely used and well-established relational database systems, academic research in Ukraine rarely focuses on them in depth—especially in direct comparison. MS SQL is often examined in the context of Microsoft technologies, while PostgreSQL typically appears in student projects or applied systems development due to its open-source nature.

Although there are broader studies evaluating relational and non-relational databases under various workloads, detailed academic comparisons between PostgreSQL and MS SQL Server remain limited. This suggests the need for more targeted research exploring how these systems perform under similar conditions and use cases, especially given their contrasting models—proprietary versus open-source—and widespread adoption.

2. METHODOLOGY

In our research we will compare these versions of the RDBMSes:

1. Microsoft SQL Server 2022 (and its supplied tools).
2. PostgreSQL 17.5 (and its supplied tools).

Comparisons were conducted on an office-grade x86_64 computer with Windows 10 installed. We will not list the details of our machine, as we will not conduct performance testing. Even though the operating system (OS) and processor architecture should not be a relevant detail in comparing the capabilites software, we mentioned them as MS SQL Server is not available on Linux distributions and different versiosn of the mentioned DBMSes have different support for architectures.

On the table 3.1 we present the list of properties that we will compare RDBMses with. These criterias primarily came from educational purposes: we will assess the basic functionalities, abilities, and feature of the systems; and our scope is limited to the most used features of DBMSes (we will not go into details that are suited for a specific task).

Table 3.1 – Comparison criterieas

Name	Description
Usage level	
Availability	Platforms, operating systems, architectures
Supplementary tools	Tools are supplied together with the server for accesing the RDBMS, analyzing, debugging
Management	Instances (servers), databases, schemas
Access level	
Basic SQL operations for data	Insertion, selection, modification and deletion
Basic SQL operations for database objects	Creation, alteration, deletion
Syntax and semantic level	
Data types	Syntax and semantics of types. Existence or absence of some types
Operators	Expression types for finding data
Database objects level	
Functions and procedures	Creation, syntactic and semantic differences
Triggers	Creation of triggers
Indexes	Types of indexes

3. RESULTS

On the table 4.1 we listed a quick summary of our findings. Differences and other specific moments are discussed below the table.

Table 4.1 – Comparison results

Type	Conclusion	Note
Usage level		
Availability	Different	Details written below
Supplementary tools	Similar	Crucial tools are present
Management	Almost similar	Details written below
Access level		
Basic SQL operations for data	Similar	Conforming SQL standard
Basic SQL operations for database objects	Similar	Conforming SQL standard
Syntax and semantic level		
Data types	Different	Different syntax, semantics, and capabilities
Operators	Almost similar	Most of the operators are present and similar
Database objects level		
Functions and procedures	Almost similar	Minor syntactic and semantic differences
Triggers	Almost similar	Minor syntactic and semantic differences
Indexes	Different	MS SQL has 2 types, PostgreSQL — only one

Starting from availability, PostgreSQL is available on a wide range of operating systems, including Linux, Windows, and macOS, and supports various hardware architectures such as x86, ARM, and more [12]. MS SQL Server, originally designed for Windows, has expanded its availability to Linux in recent years but still has more limited platform support overall. PostgreSQL's open-source nature and broad compatibility make it a popular choice for diverse environments [13], while MS SQL Server is often favored in Windows-centric infrastructures.

Continuing to the toolstet, both RDBMSes include official tools developed by their core teams to support database administration, development, and maintenance. These tools offer essential functions such as query execution, performance monitoring, and schema design. MS SQL Server provides a tightly integrated, GUI-focused environment, while PostgreSQL offers a more modular set of cross-platform utilities suited to a wide range of workflows [14]. The tools listed in table 4.2 are those commonly included with the official installers of each system.

Table 4.2. – Official Tools Included with Core Installers

Tool	Description
MS SQL	
SQL Server	The main database engine
SQL Server Management Studio	Graphical User Interface (GUI) for managing the server, databases, and writing queries
SQL Server Data Tools	Extension for Visual Studio for database development and deployment
sqlcmd	Command Line Interface (CLI) tool for running Transact-SQL (T-SQL) queries and scripts
PostgreSQL	
PostgreSQL Server	The main database engine
pgAdmin	The official GUI for managing PostgreSQL databases and running SQL queries
Stack Builder	A tool bundled with the installer that helps download and install additional PostgreSQL components and extensions
psql	The standard command-line interface for interacting with PostgreSQL databases.

As we can see, all systems provide essential tools for developing, managing, and maintaining databases. Although, MS SQL Server is closely integrated with the Microsoft ecosystem, including support for Visual Studio, which could be a competitive advantage over PotsgreSQL. However, PostgreSQL, on the other hand, takes a different approach, focusing on modularity and relying on a wide range of external tools such as DBeaver, DataGrip, pgcli, PostgREST, and various language-specific libraries.

Both PostgreSQL and MS SQL allow a single server instance to host multiple databases. While this setup appears similar, there is a key difference: MS SQL supports cross-database queries within the same instance, whereas PostgreSQL treats each database as fully isolated and does not support cross-database joins without quirks.

In both RDBMses, a database can contain multiple schemas that act as namespaces for organizing objects. However, PostgreSQL emphasizes schemas as flexible organizational tools, supporting a search path to control schema resolution. In contrast, MS SQL ties schemas more closely to ownership and permissions, without a search path mechanism. Default schems for Postgres is public, and for MS SQL is dbo.

However, crucial namespacing differences appear on a lower level — code level. In MS SQL, a function can have a parameter with the same name as a column in a table. To distinguish between them, different syntax is used: square brackets [var] refer to a column, while @var refers to a parameter. PostgreSQL does not have this distinction. If a parameter shares a name with a table column, you must explicitly prefix the column with the table name, like table.var.

The creation and manipulation of data and database objects in both MS SQL Server and PostgreSQL are largely similar, as both conform to the SQL standard. Data Definition Language (DDL) commands like creating tables, indexes, and constraints follow comparable syntax in both systems. Likewise, Data Manipulation Language (DML) operations such as inserting, updating, and deleting data also use similar SQL statements.

Each RDBMS provide data types to represent common concepts such as strings, numbers, dates, and more. However, the exact names, syntax, and sometimes the semantics of these types differ between the two systems. The table 4.3 summarizes typical data types and their equivalents in each database [6], [1], [14], [15].

Table 4.3. – Naming of common data types in PostgreSQL and MS SQL

Concept	**PostgreSQL Types**	**MS SQL Types**
String or text	VARCHAR(n), TEXT	VARCHAR(n), NVARCHAR(n), TEXT
Integer	SMALLINT, INTEGER, BIGINT	TINYINT, SMALLINT, INT, BIGINT
Boolean	BOOLEAN	BIT
Date and time	DATE, TIMESTAMP, TIMESTAMPTZ, TIME	DATE, DATETIME DATETIMETZ, SMALLDATETIME, TIME
Binary	BYTEA	VARBINARY, IMAGE
Unique indetifier	UNIQUEIDENTIFIER	UUID

In MS SQL Server, VARCHAR represents variable-length non-Unicode strings, while NVARCHAR stores variable-length Unicode strings. PostgreSQL's VARCHAR and TEXT types support Unicode by default, so there is no separate NVARCHAR type.

Regarding binary data, MS SQL Server uses types such as BINARY, VARBINARY, and the deprecated IMAGE for storing fixed-length and variable-length binary data. PostgreSQL uses a single binary data type called BYTEA (byte array) to store binary large objects (BLOBs).

Operators in PostgreSQL and MS SQL Server share many similarities due to their SQL foundations, but there are notable differences in syntax and available functionality that affect how certain operations are performed.

For example, string concatenation uses different operators: PostgreSQL uses ||, while MS SQL Server relies on the + operator. Both systems support standard logical operators such as AND, OR, and NOT. However, their bitwise operators differ slightly, with MS SQL Server using &, |, and ^, and PostgreSQL using &, |, and #. In pattern matching, PostgreSQL provides the ILIKE operator for case-insensitive matching, whereas MS SQL Server achieves this through the LIKE operator combined with collation settings.

PostgreSQL offers some distinctive features not found in MS SQL Server, including native array operators (@>, <@, &&) for working with array data types, and a rich set of JSON operators (->, ->>, #>, #>>) that simplify querying and manipulating JSON data directly within SQL queries.

Both PostgreSQL and MS SQL Server support user-defined functions for encapsulating reusable logic. In both systems, functions are declared with input parameters, a return type, and a body enclosed in BEGIN and END. They can be used in queries, procedures, and triggers for calculations or transformations.

A key difference is in the declaration details. PostgreSQL requires specifying the language with a LANGUAGE clause (commonly plpgsql), due to its support for multiple procedural languages. It also allows named and default parameters, as well as function overloading. MS SQL Server assumes T-SQL, does not support overloading, and has stricter rules for parameters.

Function types also vary. PostgreSQL uses a unified CREATE FUNCTION syntax for returning scalars, sets, or records. MS SQL Server distinguishes between scalar and table-valued functions with different structures. Additionally, in MS SQL Server, stored procedures can return output parameters, allowing them to pass values back to the caller, whereas in PostgreSQL, procedures do not return values directly—only functions can return results, and procedures are used purely for executing commands.

Within triggers, the special variables representing the modified row differ in naming conventions between MS SQL and PostgreSQL. Each system uses distinct identifiers to refer to the row data before and after the triggering event. PostgreSQL uses simplified names: NEW and OLD. MS SQL uses more verbose: INSERTED, UDPATED, DELETED.

The process of creating triggers also varies. In MS SQL, a trigger can be created with a single statement, however, PostgreSQL requires a two-step approach: first, a function that defines the trigger logic must be created, and then a trigger must be created that references this function. This difference makes trigger creation in PostgreSQL slightly more verbose.

The last crucial difference between the RDBMSes are types of indexes supported. MS SQL provides both clustered and non-clustered indexes. PostgreSQL, by contrast, offers a single type of index structure, with additional options like B-tree, hash, and others, but the concept of clustered versus non-clustered indexes as in MS SQL is not present.

4. CONSLUSIONS

This paper presents a detailed comparison of two major database systems: Microsoft SQL Server and PostgreSQL. We looked at these systems from multiple angles, including where they can run, what tools they offer, how they handle data, and their programming languages.

The two systems have very different backgrounds. Microsoft SQL Server started in 1989 as a commercial product built on Sybase technology. Microsoft has kept developing it as a paid, proprietary system that works well with other Microsoft products. PostgreSQL began in 1995 from university research at UC Berkeley. It grew out of the earlier

Postgres project and became a free, open-source system maintained by volunteers worldwide. This difference in origins - commercial versus academic - has influenced how each system developed and who uses them.

Our literature review shows that few academic studies directly compare PostgreSQL and MS SQL Server, particularly within the Ukrainian research context. Most papers that reference these databases employ them as tools for other research rather than examining the databases themselves. We identified some valuable studies on PostgreSQL's transaction handling and broader comparisons of database performance, but direct comparisons between these two systems remain limited.

We established a systematic framework to compare the databases across four main areas: usage patterns, access methods, programming syntax, and database features. This approach enabled us to build a comprehensive comparison covering platform support, available tools, data handling, SQL compliance, data types, operators, and advanced features like functions and triggers.

Our comparison revealed many similarities between the two systems. Both follow SQL standards well and handle basic data operations in similar ways. Both offer comprehensive tools for database management, including graphical interfaces and command-line options. They also both support important features like custom functions, stored procedures, triggers, and multiple databases on one server.

The differences reflect their distinct origins and target users. PostgreSQL runs on more operating systems and hardware types because it's open source, while SQL Server was traditionally Windows-focused (though it now supports Linux too). The systems handle data types differently, especially for text encoding and binary data. They also use different syntax for some operations - PostgreSQL has unique features like built-in array and JSON support, while SQL Server offers different indexing options.

Based on our analysis, we recommend SQL Server for environments where Microsoft products are already established, as it integrates smoothly with that ecosystem. For most other applications, PostgreSQL offers better value due to its free licensing, broader platform support, and extensive capabilities. From an educational perspective, we strongly advocate for integrating PostgreSQL and other open-source database systems into university curricula. This integration would not only provide students with industry-standard alternatives but also reduce institutional costs. Given the underrepresentation of open-source alternatives in many Ukrainian universities, incorporating PostgreSQL into database courses would prepare students for the technology variety they will encounter in their professional careers.

This study adds to the small amount of academic research comparing these important database systems in Ukrainian science. While we focused on qualitative comparison, future work should include performance testing and benchmarking to provide quantitative data alongside our findings.

References

[1] A. Jorgensen, J. Segarra, P. LeBlanc, J. Chinchilla, and A. Nelson, *Microsoft SQL server 2012 bible*. John Wiley & Sons, 2012.

[2] "Release Notes." [Online]. Available: https://www.postgresql.org/docs/release/0.01/

[3] Microsoft, "Editions and supported features of SQL Server 2022." Accessed: Jun. 08, 2025. [Online]. Available: https://learn.microsoft.com/en-us/sql/sql-server/editions-and-components-of-sql-server-2022?view=sql-server-ver17

[4] Ірина Шаклеіна, Надія Ших, and Роман Білий, "Опрацювання баз даних засобами MS Visual Studio," 2018.

[5] О. М. Верес, В. Л. Мельник, and Л. Б. Чирун, "Застосування MS SQL Server 2005 для побудови інтелектуальної складової інформаційної системи," 2008.

[6] B. Momjian, *PostgreSQL: introduction and concepts*, vol. 192. Addison-Wesley New York, 2001.

[7] Тарас Чмир and Назарій Євгенович Бурак, "Проєктування інформаційної системи обліку обслуговування та використання транспортних засобів на базі сховища даних PostgreSQL," 2024.

[8] Сергій Сергійович Вакуленко, "Дослідження та обгрунтування оптимального типу бази даних для побудови OLAP репозиторію для сайту-агрегатора пошуку роботи," 2024.

[9] Ю. Р. Глинка and Р. Б. Вовк, "МЕХАНІЗМИ ЗАБЕЗПЕЧЕННЯ НАДІЙНОСТІ ТРАНЗАКЦІЙ У СИСТЕМІ КЕРУВАННЯ БАЗАМИ ДАНИХ POSTGRESQL," *Інформаційні технології і автоматизація–2024/Матеріали XVII міжнародної науково-практичної конференції. Одеса, 31 жовтня-1 листопада 2024 р.-Одеса, Видавництво ОНТУ, 2024 р.–847 с. Збірник включає матеріали доповідей учасників конференції, які об'єднані за тематичними напрямками конференції.*, p. 178, 2024.

[10] СО Скрипка and СВ Шаров, "ПОРІВНЯННЯ РЕЛЯЦІЙНИХ БАЗ ДАНИХ ДЛЯ ВИКОРИСТАННЯ В ІНФОРМАЦІЙНИХ СИСТЕМАХ," *Українські студії в європейському контексті: зб. наук. пр. 2024. № 8. 382 с.*, p. 284, 2024.

[12] T. P. G. D. Group, "PostgreSQL 17.5 Documentation." Accessed: Jun. 08, 2025. [Online]. Available: https://www.postgresql.org/docs/17/index.html

[14] D. Fontaine, *Mastering PostgreSQL in Application Development*. Lulu.com, 2017. [Online]. Available: https://books.google.com.ua/books?id=9ZDIswEACAAJ

[15] R. Mistry and S. Misner, *Introducing Microsoft SQL Server 2014*. Microsoft Press, 2014.

Отримана в редакції 12.06.2025. Прийнята до друку 18.06.2025. Received 12 June 2025. Approved 18 June 2025. Available in Internet 30 June 2025