DOI: 10.31319/2519-8106.1(52)2025.325957
UDC 004.657

Yalova Kateryna[1], Candidate of Technical Sciences, Associate Professor, Head of the Department of the Systems software
Ялова К.М., кандидат технічних наук, доцент, кафедра програмного забезпечення систем
ORCID: 0000-0002-2687-5863
e-mail: yalovakateryna@gmail.com

Ismailov Vitaliy[1], PhD student, Department of the Systems software
Ісмаілов В.В., аспірант кафедри програмного забезпечення систем
email: ivv15081996@gmail.com

Babenko Mykhailo[1], Candidate of Technical Sciences, Associate Professor, Associate Professor of the Department of the Systems software
Бабенко М.В., кандидат технічних наук, доцент, кафедра програмного забезпечення систем
ORCID: 0000-0003-1013-9383
e-mail: mvbab@ukr.net

Poliakova Nataliia[2,3], Candidate of Technical Sciences, Associate Professor of the Department of Electronics, information systems and software
Полякова Н.П., кандидат технічних наук, доцент, кафедра електроніки, інформаційних систем та програмного забезпечення
ORCID: 0000-0002-1252-4149
e-mail: natalypol47@gmail.com

[1]Dniprovsky State Technical University, Kamianske
Дніпровський державний технічний університет, м. Кам’янське
[2]Запорізький національний університет, м. Запоріжжя
Zaporizhzhia National University, Zaporizhzhia
[3]Університет інформаційних технологій та менеджменту, м. Ряшів, Польща
University of Information Technology and Management, Rzeszow, Poland

NUMERICAL EVALUATION OF DATA MODEL PERFORMANCE FOR MULTIDIMENSIONAL DATA ANALYSIS

ЧИСЕЛЬНЕ ОЦІНЮВАННЯ ШВИДКОДІЇ МОДЕЛЕЙ ДАНИХ В АНАЛІЗІ БАГАТОВИМІРНИХ ДАНИХ

The article is dedicated to the numerical evaluation of the efficiency of two data models: the star and snowflake schemas. The study presents the results of designing these models, including a detailed description of their dimensions, facts, and values. For the research, a dataset comprising the results of the Ukrainian National Multisubject Test from 2022 to 2024 was used. The numerical evaluation employed performance and data redundancy, and storage space using as key metrics. For each analytical operation — including slicing, drill-down, dicing, roll-up, and pivoting — the performance assessment of the developed models was presented. Performance evaluation was conducted automatically using SQL Server Profiler, while data redundancy was measured based on the Data Storage Overhead metric. The numerical evaluation demonstrated that the star schema performs drill-down and roll-up operations 36 % and 70 % faster, respectively, than the snowflake schema, while exhibiting 33 % higher data redundancy.

***Keywords**: multidimentional data analysis; hypercubes; snowflake data scheme; star data scheme; analytic operations.*

Статтю присвячено чисельному оцінюванню ефективності двох моделей даних, що застосовуються в аналітичних системах управління даними: «зірка» та «сніжинка». Представлено результати проєктування цих моделей із детальним описом їхньої структури, включаючи виміри, факти та значення. Для досягнення мети дослідження виконано такі завдання: проєктування моделей даних у межах вибраної предметної області; їхнє розгортання у вигляді OLAP-гіперкуба; реалізація основних аналітичних операцій багатовимірного аналізу даних із фіксацією швидкодії; проведення порівняльного аналізу отриманих результатів і формулювання висновків щодо ефективності кожної моделі.

Для проведення чисельного дослідження використовувався масив результатів складання Національного мультипредметного тесту України за 2022—2024 роки, оскільки ці дані мають багатовимірну природу, поєднуючи атрибути з різних незалежних сутностей предметної області. Для чисельного оцінювання застосовувалися такі метрики: швидкодія виконання запитів, надмірність даних та ефективність використання пам'яті.

У роботі оцінено швидкодію виконання базових аналітичних операцій — зрізу, створення підкубу, агрегації, деталізації, та обертання — шляхом автоматизованого вимірювання часу обробки запитів у середовищі SQL Server Profiler. Для кожної операції надано результати тестування та приклади трьох виконаних запитів. Аналіз даних щодо швидкодії моделей показав, що операція зрізу виконувалася на 2,86 % повільніше на моделі «сніжинка», тоді як створення підкубів і обертання — на 37 % та 16,37 % відповідно. Найбільша різниця у швидкодії у 70,47 % зафіксована для операції деталізації. Єдина операція, де модель «сніжинка» переважала за швидкодією, — агрегація (2,39 % швидше), що пояснюється її нормалізованою структурою та меншою надмірністю даних. Отримані результати представлено у вигляді гістограм.

Надмірність даних визначалася за метрикою Data Storage Overhead. Попри збільшення кількості таблиць у 1,83 рази, модель «сніжинка» містила лише на 0,5 % більше записів завдяки її нормалізованій структурі. Водночас значення DSO для модель «зірка» склала 33 %, що пояснюється значним дублюванням даних у ненормалізованих таблицях вимірів.

***Ключові слова**: багатовимірний аналіз даних; гіперкуб; модель даних «зірка»; модель даних «сніжинка»; аналітичні операції.*

Problem's formulation

Various data models are used for data storage in information systems, defining the organization, storage format, and data processing methods. The most common types of systems designed for data storage include databases, particularly relational databases and data warehouses. If an information system is intended for daily transaction processing operations, such as data insertion, updating, and deletion, it is more appropriate to use relational databases for data storage. In the case of developing business intelligence (BI) systems that require comprehensive analysis of multidimensional or big data for decision-making, data warehouses (DW) should be employed. This is because they typically contain historical data that is updated periodically and used for trend analysis and decision-making.

For the design and development of efficient data storage structures, particularly in Online Analytical Processing (OLAP) systems, various data models are employed, the most common of which are the star schema (StS) and snowflake schema (SnS) [1]. StS is used in DWs for organizing multidimensional data, where a central fact table is linked to multiple dimension tables. This structure ensures query simplicity and fast data access. SnS is an extension of the StS, in which dimension tables are normalized, meaning they are divided into sub-tables. This approach reduces data redundancy but complicates query structures. The choice of data model significantly impacts performance, scalability, and the efficiency of multidimensional data analysis. Given the continuous growth in data volume and complexity, determining the optimal model is critically important for ensuring fast and accurate analytical processing [2].

The performance of implemented data models depends on several factors, including the degree of normalization, data redundancy, indexing mechanisms, and query execution characteristics. The efficiency of OLAP operations such as slicing, dicing, roll-up, drill-down, and pivot directly impacts the overall performance of analytical and BI systems. Therefore, numerical evaluation of the performance of different data models is crucial for determining their suitability for processing large-scale multidimensional data.

Analysis of recent research and publications

In modern BI systems, DWs play a key role in enabling efficient multidimensional data analysis. Various data models, such as the StS and SnS, are widely used for structuring DWS, and their effectiveness has been the subject of numerous studies.

In the late 1990s, Ralph Kimball was the first to formalize the concept of the StS, advocating it as the primary model for DW design. He described it as a simple and fast-to-implement structure, optimized for multidimensional analysis. The SnS, on the other hand, was developed as an extension of the StS to accommodate hierarchical structures within dimension tables. One of the first researchers to advance the idea of data normalization for designing the SnS was Bill Inmon.

Studies [3—5] emphasize the importance of selecting the optimal data model in the context of query performance and data storage efficiency. Research on OLAP query performance across different data models [6, 7] indicates that the execution time of analytical operations (slicing, dicing, roll-up, drill-down, pivot) is highly dependent on the chosen data schema. Additionally, recent studies [1, 8] suggest that data model performance is significantly influenced by the software implementation of the DWs, the indexing mechanisms employed, and the hardware resources available.

Despite the substantial body of existing research, the issue of comprehensive numerical evaluation of data model performance in the context of multidimensional analysis remains highly relevant. Therefore, further investigation in this area is crucial for optimizing DW performance and enhancing the efficiency of analytical and BI systems.

Formulation of the study purpose

The aim of this study is to present the results of a numerical evaluation of two data models — StS and SnS — which are used for the software implementation of DWs designed for multidimensional data analysis. To achieve this goal, the study includes the following tasks:

- designing data models within the chosen domain;
- deploying these models as an OLAP data hypercube;
- executing key analytical operations for multidimensional data analysis while recording their performance;
- conducting a comparative analysis of the obtained results;
- formulating conclusions regarding the efficiency of the developed data models.

The results of this research provide a well-founded basis for determining the optimal data model for multidimensional data analysis, depending on key criteria such as performance, memory efficiency, and data redundancy reduction. These factors are critically important for building efficient DWs.

Presenting main materials

A data model is a formalized representation of the structure, relationships, and constraints of data, used for storage, processing, and analysis. It defines how data is organized, interconnected, and accessed. Data models ensure that data is stored and structured according to specific principles. The two most common data models used for structuring DWs are the StS and SnS [9]. Regardless of the model type, their architecture consists of fact tables, dimension tables, and a set of measures. The primary distinction between the StS and SnS lies in the relationship structure between the fact table and the dimension tables. In the StS, all dimensions are directly connected to the fact table, and the data within these tables is denormalized or only weakly normalized. In contrast, the SnS allows dimension tables to contain hierarchies, represented by child tables that emerge through the data normalization process.

Suppose that F represents the fact table, $M=\{m_1,\ldots,m_k\}$ is the set of measures in the fact table, and $K=\{k_{d1},\ldots,k_{dn}\}$ is the set of foreign keys linking the fact to the i-th dimension. Then, the fact table for both StS and SnS can be described as:

$$F = \langle ID_F, K, M \rangle, \tag{1}$$

where ID_F — is the primary key of the fact table.

Each record in the fact table is uniquely associated with the corresponding records in the dimension tables through a functional dependency, which implies that each combination of foreign keys determines a set of measures in the fact table, described as:

$$\left(k_{d1},\ldots,k_{dn}\right) \longrightarrow \left(m_1,\ldots,m_k\right). \quad (2)$$

If $D=\{d_1,\ldots,d_n\}$ represents the set of dimension tables, and $A=\{a_{i1},\ldots,a_{im}\}$ is the set of attributes for each i-th dimension, then the dimension table for the StS can be described as:

$$d_i = \left\langle ID_{d_i}, A \right\rangle, \quad i = 1,.., n, \quad (3)$$

where n is the number of dimension tables, and, ID_{di} is the primary key of the i-th dimension table.

For mathematical description of the dimension tables in the SnS, it is necessary to account for its hierarchical structure [10] by introducing the set of foreign keys $FK=\{fk_{s1},\ldots,fk_{im}\}$ and the set of normalized sub-dimension tables $S=\{s_{11},\ldots,s_{ij}\}$. Taking this into consideration, expression (3) is transformed into:

$$d_i = \left\langle ID_{d_i}, FK, A \right\rangle. \quad (4)$$

In this case, each s_{ij} normalized child dimension table can be described as:

$$s_{ij} = \left\langle ID_{s_{ij}}, A \right\rangle, \quad (5)$$

where $ID_{s_{ij}}$ — is the primary key of the s_{ij} normalized child dimension table.

The relationship between the fact table and the dimension tables can be mathematically described through the set of foreign keys as follows:

$$\forall_i \in \{1,\ldots,n\},\ k_{d_i} \in d_i,\ \ \exists f \in F : f\left\lfloor k_{d_i} \right\rfloor = d\left\lfloor k_{d_i} \right\rfloor. \quad (6)$$

Expression (6) shows that each value of the foreign key in the fact table refers to the corresponding record in the related dimension table.

For the numerical study of the StS and SnS, a dataset of results from the National Multi-Subject Test (NMT) of Ukraine for the years 2022-2024 was used [11]. The NMT is a test structured in a 3+1 format, where three subjects are mandatory, and one subject is chosen by the graduate based on personal preference. The results of the NMT have a multidimensional nature, as they combine data from various independent entities within the data domain. Tabl. 1—2 describe the composition of dimensions and metrics of the data domain, which were implemented in the developed StS and SnS.

Table 1. Composition of the domain dimensions

Dimension	Name of a dimension	Description of a dimension
d_1	Time dimension	Represents the years in which the test was conducted
d_2	Location dimension	Describes the territorial components related to test participants
d_3	Course dimension	Lists the academic disciplines available for students to take as part of the test
d_4	Participants dimension	Includes quantitative and categorical attributes of students who took the test
d_5	School dimension	Describes the types and attributes of the schools where students studied
d_6	Status dimension	Represents the classification and status of the test outcomes

Table 2. Composition of the domain measures

Measure	Description of a measure
m_1	Total count of students who registered for the test
m_2	Count of students who attempted the test but did not pass
m_3	The mean score across all test takers
m_4	Students who achieved a perfect score in at least one subject
m_5	Students who achieved a perfect score in two subjects
m_6	Students who achieved a perfect score in three subjects
m_7	Students who obtained the highest possible total score

The models were programmatically implemented using SQL Analysis Services, and their diagrams are presented in Fig. 1 and 2, respectively.

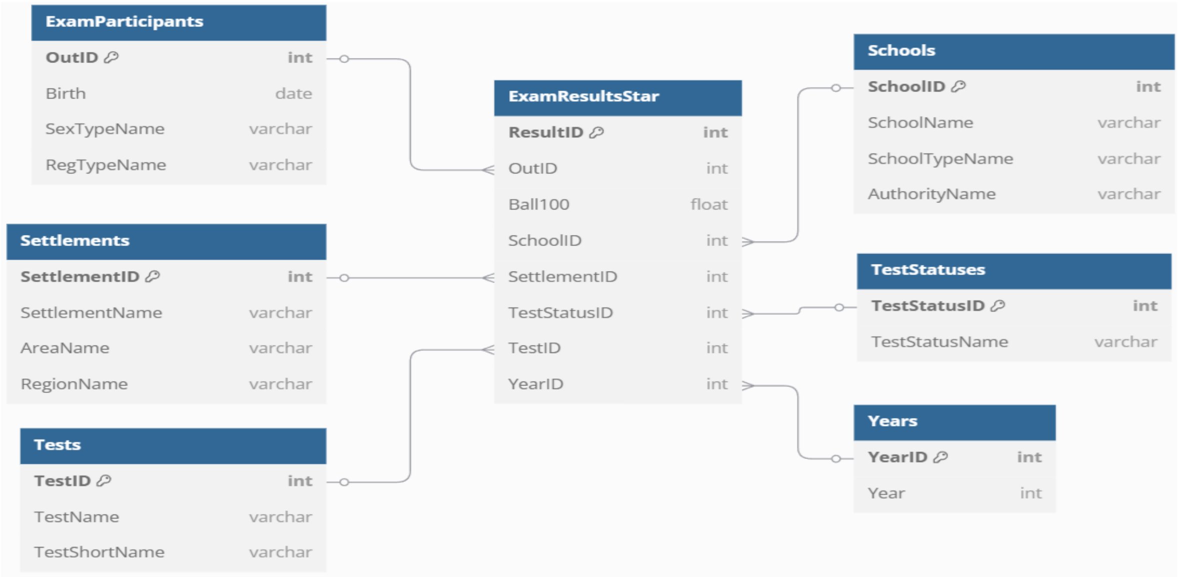

Fig. 1. Star schema diagram

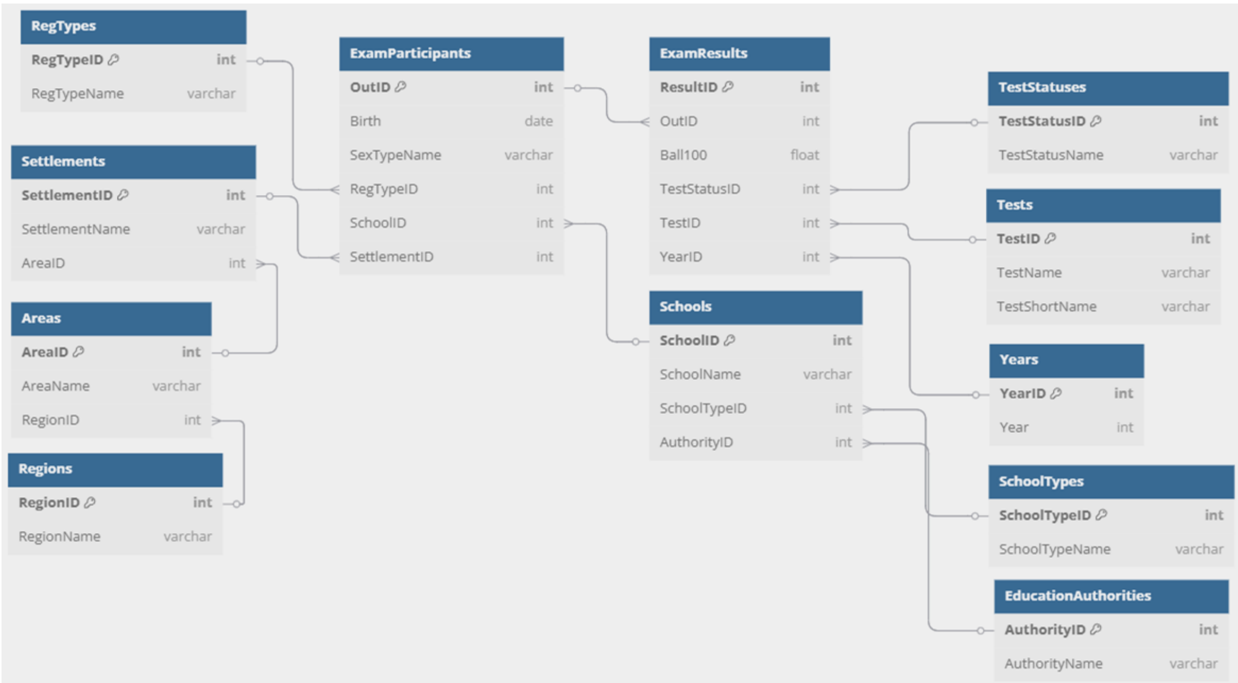

Fig. 2. Snowflake schema diagram

To conduct a numerical comparison of the developed models, they were deployed as a data hypercube using Microsoft SQL Server Analysis Services, consisting of six dimensions. The set of dimensions D was normalized to the third normal form (3NF) for the SnS. This normalization ensured the atomicity of attributes in the fact tables, eliminated transitive dependencies between their fields, and guaranteed that all attributes depended solely on the primary key. The dimensions d_2, d_4, d_5 were represented with hierarchical relationships between their components. In the case of the location dimension, it is defined as: $d_2=\{d_{2,1}, d_{2,2}, d_{2,3}\}$, where $d_{2,1}\to d_{2,2}\to d_{2,3}$, This hierarchy represents the administrative division: region → district → settlement. For the participants dimension, defined where: $d_4=\{d_{4,1}, d_{4,2}\}$, де $d_{4,1}\to d_{4,2}$, the hierarchy reflects the relationship: registration type → test participant.

The school dimension is described as: $d_5=\{d_{5,1}, d_{5,2}, d_{5,3}\}$, where $d_{5,1}\to d_{5,2}\to d_{5,3}$, to represent the hierarchical structure: education department → type of educational institution → specific educational institution.

Thus, the StS consists of one fact table and six dimension tables, while the SnS consists of one fact table and eleven dimension tables. The fact table in both models contains 2 891 8146 records, whereas the dimension tables contain 875 044 records for the StS and 878 747 records for the SnS. This indicates that doubling the number of dimension tables in the SnS led to only a 0.5 % increase in stored records.

The total memory usage after populating the tables with data was 549,13 MB for the StS and 412,72 MB for the SnS. Although the StS contained 0.5 % fewer records, it required 33 % more memory compared to the SnS. This is due to the significant data redundancy in its denormalized structure. If we assume that the memory size S_m occupied by the SnS represents the data volume without redundancy, and S_t represents the memory size of the StS, then the Data Storage Overhead (DSO) metric can be used to evaluate the storage efficiency of both models:

$$DSO = \frac{S_t - S_m}{S_m} \times 100\% . \tag{7}$$

This allows us to conclude that ≈33 % of the storage space in the StS is occupied by duplicated data.

To evaluate the performance of the developed models, a series of analytical operations was conducted, executing 50 queries for each of the following operations: slicing, dicing, roll-up, drill-down, and pivoting. The execution speed of the result set retrieval was automatically recorded. Tabl. 3—6 present three sample queries for each operation.

Let the developed data hypercube G be described as:

$$G = \langle D, M \rangle , \tag{8}$$

where $D = \{d_1,\ldots,d_6\}$ — the set of dimensions of the hypercube, corresponding to the set of dimension tables in each data model; $M=\{m_1,\ldots,m_7\}$ represents the set of measures within the hypercube, characterizing the process as described in Tabl. 1 and 2, respectively.

Thus, the slicing operation, which fixes the value of a dimension d_j at a specific value d_j^* and reduces the dimensionality of the data hypercube, results in the creation of a data slice, which can be expressed as:

$$G' = \{m_k \mid d_j = d_j^{*}, \forall d_i \in D \setminus \{d_j\}\} , \tag{9}$$

where G′ — is the new hypercube after fixing the dimension.

Tabl. 3 presents examples of queries for performing the slicing operation and the results of their evaluation.

Table 3. Examples of queries for executing the slicing operation

Query	Measure	Slicing parameter	Data scheme	Resulting set, records	Query execution time, ms
Q1	m_1	d_1=2023	Star scheme	886 805	286
			Snowflake scheme		294
Q2	m_1	d_3=enrolled	Star scheme	2 524 549	744
			Snowflake scheme		738
Q3	m_1	d_6=math	Star scheme	1 997 463	704
			Snowflake scheme		712

The dicing operation returns a subset of values across multiple dimensions. If dicing is performed for dimensions d_i and d_j, the new data hypercube G' is described as:

$$G' = \{m_k \mid d_i \in D_i^*, d_j \in D_j^*, \forall d_n \in D \setminus \{d_i, d_j\}\} , \tag{10}$$

where $D_i^* \subseteq d_i$ and $D_j^* \subseteq d_j$ — are subsets of the dimensions.

Tabl. 4 presents examples of queries for performing the dicing operation and their evaluation results.

Table 4. Examples of queries for executing the dicing operation

Query	Measure	Slicing parameter	Data scheme	Resulting set, records	Query execution time, ms
Q1	m_1	d_1=2023 and $d_{2,1}$= Dnipropetrovsk region	Star scheme	105 496	27
			Snowflake scheme		39
Q2	m_1	d_1=2024 and d_6=math	Star scheme	234 112	27
			Snowflake scheme		36
Q3	m_1	$d_{2,1}$= Dnipropetrovsk region and $d_{5,2}$=gymnasium and d_1=2022	Star scheme	101 207	37
			Snowflake scheme		32

The roll-up operation, which performs data aggregation by ascending a level in the hierarchy of a dimension for the metric m_k aggregated at level $d_{i,k}$, is mathematically described as:

$$m_k' = \sum_{d_{i,k+1}} m_k, \quad d_{i,k+1} \in d_i, \tag{11}$$

where $d_{i,k+1}$ represents the next level in the hierarchy being aggregated.

Examples of queries for performing the roll-up operation and their evaluation results are presented in the Tabl. 5.

Table 5. Examples of queries for executing the roll-up operation

Query	Measure	Slicing parameter	Data scheme	Resulting set, records	Query execution time, ms
Q1	m_1	d_2= $d_{2,1}$→$d_{2,2}$→ $d_{2,3}$	Star scheme	14894	646
			Snowflake scheme		653
Q2	m_3	d_5= $d_{5,1}$→$d_{5,2}$→ $d_{5,3}$	Star scheme	341	290
			Snowflake scheme		365
Q3	m_2	d_4= $d_{4,1}$, $d_{4,2}$	Star scheme	25	47
			Snowflake scheme		45

The drill-down operation, which is performed by descending to a lower level in the hierarchy for the metric $m_{i,k}$ aggregated at level $d_{i,k}$, can be mathematically described as:

$$m_{d_{i,k+1}} = m_{d_{i.k}}, \text{ where } d_i \downarrow d_{i+1}. \tag{12}$$

Tabl. 6 presents examples of queries for performing the drill-down operations.

Table 6. Examples of queries for executing the drill-down operation

Query	Measure	Slicing parameter	Data scheme	Resulting set, records	Query execution time, ms
Q1	m_3	$d_{2,3} = d_{2,2}\downarrow$	Star scheme	310 466	302
			Snowflake scheme		261
Q2	m_1	$d_{2,3} = d_{2,2}\downarrow$	Star scheme	191 056	218
			Snowflake scheme		173
Q3	m_2	$d_{2,2} = d_{2,1}\downarrow$	Star scheme	504	154
			Snowflake scheme		334

The pivot operation changes the way dimensions are displayed in the hypercube. After applying the permutation operator P between two dimensions d_i and d_j the data hypercube G' is described as:

$$G' = P(M, d_i, d_j). \tag{13}$$

The histograms with grouping, shown in Fig. 3 and 4, display comparative information regarding the performance of the developed data models.

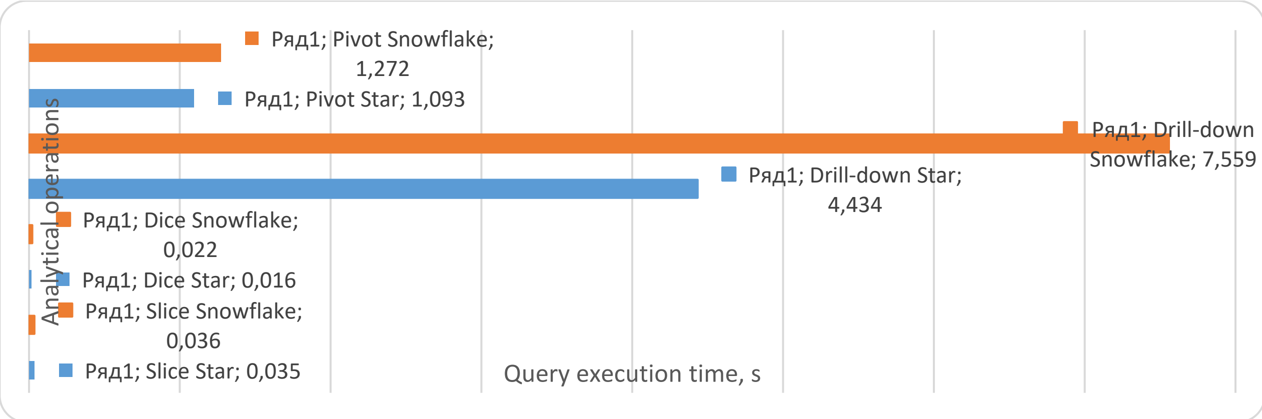

Fig. 3. Comparison of the performance of the developed models for the operations of slicing, dicing, drill-down, and pivot

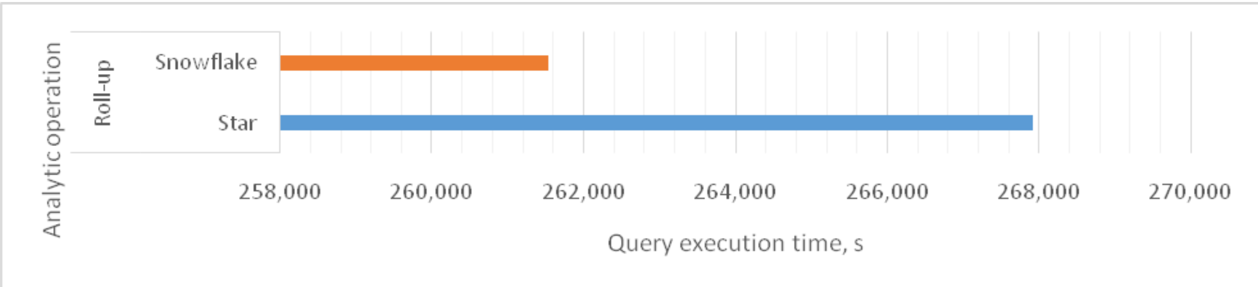

Fig. 4. Comparison of the performance of the developed models for the roll-up operation

To generalize the performance indicators of the models, the time spent on generating the resulting datasets was normalized to the average query execution speed, defined as the time required to return 100 000 records from the resulting dataset for all analytical operations of the hypercube.

The analysis of the data presented in Fig. 3 and 4 shows that for the slicing operation, the performance values differ insignificantly, with the slicing operation taking 2,86 % longer on the SnS. In the case of the dicing and pivoting operations, the performance difference becomes more significant, with these operations taking 37 % and 16,37 % longer, respectively, on the SnS. The greatest performance difference, which amounted to 70,47 %, was demonstrated during the drill-down operation. The only operation where the SnS outperformed in terms of performance was roll-up, which was performed 2,39 % faster, which may be explained by the lower data redundancy and its normalized dimension structure. Overall, the StS demonstrates higher performance for most analytical queries, especially in the dicing and drill-down operations.

By evaluating the performance and memory usage criteria, a comparative characterization of the developed models was created, with the main values presented in Tabl. 7.

Table 7. Comparative characteristics of the StS and SnS

Characteristic	Star Schema	Snowflake Schema
Structure Simplicity	Higher	Low (due to data normalization)
Query Speed	Higher (less JOIN operations)	Lower (more JOIN operations)

Continue of the table 7

Number of JOIN operations	O(n)	O(n+m)
Data Redundancy	High	Low
Data Storage Volume	Bigger	Smaller
Ease of Modification	It is more difficult to update due to data duplication.	Easier to update due to data distribution

As seen from the data presented in Tabl. 7, the StS features a high structural simplicity and faster query performance due to fewer JOIN operations. However, it has high data redundancy and a larger data storage volume. The SnS, on the other hand, is characterized by greater structural complexity due to normalization, which results in slower query performance due to an increased number of JOIN operations. It exhibits lower data redundancy and requires less storage, while being more convenient for modifications due to the distribution of data. It should be noted that all test queries were executed on non-indexed data. Indexing the data in the SnS can be utilized to improve its performance.

Conclusions

The numerical evaluation showed that the StS is more optimal in terms of performance for most OLAP operations, except for roll-up. This is because in the StS, all dimension attributes are located in a single table, which avoids additional joins when executing SQL queries. Additionally, it has a simple structure and minimizes the number of necessary joins between tables. However, the SnS model may be more appropriate if the priority is space efficiency and structural clarity of data, especially for large DWs with many hierarchy levels, as it ensures table normalization and reduces data redundancy.

The evaluation of the size and structure of the created models led to the following conclusions. Despite the increase in the number of tables by a factor of 1,83, the SnS model contains only 0,5 % more records, which is due to its normalized structure. At the same time, the StS model occupies 33 % more memory, which is explained by significant data duplication in the denormalized dimension tables. Therefore, in terms of efficient memory usage, the SnS model is more optimal, as its normalization significantly reduces excessive data storage.

Based on the evaluation of performance, generalized recommendations were formulated regarding the selection of a data model for performing analytical multidimensional operations, specifically:

– if the system executes frequent drill-down, dicing, or pivot operations, the StS model is preferred;

– if the focus is on aggregated reports and reducing data redundancy, the SnS model may be more effective.

The results of the numerical study presented in this work can be used to choose the optimal DW schema depending on specific business needs and the volume of analytical queries for other data domains.

References

[1] Mohammed, K.I. (2019). Data Warehouse Design and Implementation Based on Star Schema vs. Snowflake Schema. International Journal of Academic Research in Business and Social Sciences, 7(5), 25—38. doi: 10.6007/IJARBSS/v9-i14/6502.

[2] Rorimpandey, G. C., Sangkop, F. I., Rantung, V. P., Zwart, J. P., Liando, O. E. S., Mewengkang, A. (2018). Data model performance in data warehousing. Materials Science and Engineering, 306, 012044. doi: 10.1088/1757-899X/306/1/012044.

[3] Kossman, J., Papenbrock, T., Maumann, F. (2021). Data dependiecies for query optimization: a survey. The International Journal of Very Large Data Bases, 31 (1), 1—22. doi: 10.1007/s00778-021-00676-3.

[4] Taipalus, T. (2025). On the effects of logical database design on database size, query complexity, query performance, and energy consumption. URL: https://arxiv.org/abs/2501.07449v1

[5] Ribeiro, A., Silva, A., Rodrigues da Silva, A. (2015). Data Modeling and Data Analytics: A Survey from a Big Data Perspective. Journal of Software Engineering and Applications, 8(12), 1—18. doi: 10.4236/jsea.2015.812058.

[6] Li, X., Shen, Q., Yang, T. (2024). Design and optimization of multidimensional data models for enhanced OLAP query performance and data analysis. Applied and Computational Engineering, 1, 161—166. doi: 10.54254/2755-2721/69/20241503.

[7] Forresi, C., Gallinucci, E., Golfarelli, M. (2021). A dataspace-based framework for OLAP analyses in a high-variety multistore. The VLDB Journal, 30, 1017—1040. doi:10.1007/s00778-021-00682-5.

[8] Abbasi, M., Bernardo, M., Vaz, P., Silva, J., Martins, P. (2024). Revisiting Database Indexing for Parallel and Accelerated Computing: A Comprehensive Study and Novel Approaches. Informaion, 15(8), 429. doi: 10.3390/info15080429.

[9] Yalova, K., Babenko, M., Ismailov, V. (2024). OLAP hypercubes as a tool for analyzing multidimensional highly structured data. Mathematical Modelling, 51, 57—65. doi: 10.31319/2519-8106.2(51)2024.317498.

[10] Azzini, A., Ceravolo, P., Colella, M. (2019). Performances of OLAP operations in graph and relational databases. Knowledge Management in Organizations, 1027, 282—293. doi: 10.1007/978-3-030-21451-7_24.

[11] Opendata: Statistics on national multi-subject testing. (2025).
URL: https://zno.testportal.com.ua/opendata.

Список використаної літератури

1. Mohammed K.I. Data Warehouse Design and Implementation Based on Star Schema vs. Snowflake Schema. *International Journal of Academic Research in Business and Social Sciences*. 2019. №7(5). P. 25—38. doi: 10.6007/IJARBSS/v9-i14/6502.
2. Rorimpandey G. C., Sangkop F. I., Rantung V. P., Zwart J. P., Liando O. E. S., Mewengkang A. Data model performance in data warehousing. *Materials Science and Engineering*. 2018. №306. ID 012044. doi: 10.1088/1757-899X/306/1/012044.
3. Kossman J., Papenbrock T., Maumann F. Data dependiecies for query optimization: a survey. *The International Journal of Very Large Data Bases*. 2021. №31(1), P. 1—22. doi: 10.1007/s00778-021-00676-3.
4. Taipalus T. On the effects of logical database design on database size, query complexity, query performance, and energy consumption. URL: https://arxiv.org/abs/2501.07449v1 (дата звернення: 10.02.2025).
5. Ribeiro A., Silva A., Rodrigues da Silva A. Data Modeling and Data Analytics: A Survey from a Big Data Perspective. *Journal of Software Engineering and Applications*. 2015. № 8(12). P. 1—18. doi: 10.4236/jsea.2015.812058.
6. Li X., Shen Q., Yang T. Design and optimization of multidimensional data models for enhanced OLAP query performance and data analysis. *Applied and Computational Engineering*. 2024. №1. P. 161—166. doi: 10.54254/2755-2721/69/20241503.
7. Forresi C., Gallinucci E., Golfarelli M. A dataspace-based framework for OLAP analyses in a high-variety multistore. *The VLDB Journal*. 2024. №30. P. 1017—1040. doi:10.1007/s00778-021-00682-5.
8. Abbasi M., Bernardo M., Vaz P., Silva J., Martins P. Revisiting Database Indexing for Parallel and Accelerated Computing: A Comprehensive Study and Novel Approaches. *Information*. 2024. №15(8). ID 429. doi: 10.3390/info15080429.
9. Yalova K., Babenko M., Ismailov V. OLAP hypercubes as a tool for analyzing multidimensional highly structured data. *Mathematical Modelling*. 2024. Vol. 51. P. 57—65. doi: 10.31319/2519-8106.2(51)2024.317498.
10. Azzini A., Ceravolo P., Colella M. Performances of OLAP operations in graph and relational databases. *Knowledge Management in Organizations*. 2019. Vol. 1027. P. 282—293. doi: 10.1007/978-3-030-21451-7_24.
11. Opendata: Statistics on national multi-subject testing. URL: https://zno.testportal.com.ua/opendata (дата звернення: 01.02.2025).

Надійшла до редколегії 11.04.2025