Optimization of the Database Structure of a Distributed Corporate Information System Node Using the Analytic Hierarchy Process

Mykhailo Dvoretskyi[a], Svitlana Dvoretska[a], Hlib Horban[a] and Yuriy Nezdoliy[a]

[a] *Petro Mohyla Black Sea National University, 68-Desantnykiv St 10, Mykolaiv, 54003, Ukraine*

Abstract

The relevance of the problem of optimizing the database structure of a node in corporate information systems (CIS) is due to the widespread use of information technologies of multi-level, geographically dispersed computer systems, including those with distributed databases.
One of the research aims is to determine and build a mathematical model of the optimality criteria for the structure of a remote node of the distributed corporate information system database. The statistics of user SQL-queries activity is taken into account and presented in the form of a multidimensional database. Criteria of the model effectiveness are formulated, which are independence from the central node of the database, the size of the local database, and an indicator of the level of need for data synchronization.
The problem of multicriteria optimization is solved by using of hierarchy analysis method. Among the using method's features can be mentioned: different sets of optimality criteria for the evolving individuals; quantifying of the data representation marker value into 5 alternatives and automatically presetting the matrices of pairwise comparisons on the last level of the hierarchy.
Solving the problem of multicriteria analysis and choosing the best alternative makes possible to determine the optimal level of the data representation marker. It makes possible to classify the attributes and tuples of DB relations according to their representation on the node of distributed CIS.

Keywords [1]

Corporate information system, database management system, distributed database, SQL-query, data replication, multidimensional analysis, multicriteria problem, analytic hierarchy process.

1. Introduction

In information systems development, there is a trend of transition from local to distributed databases (DDB). There are many database management systems (DBMS) that allow you to host, maintain and process data on various nodes of computer information systems (CIS). The main task of distributed database management systems is to provide access control to the data of many users and ensure the integrity and consistency of data [1]. Within one company there is a need to automate different types of accounting [2, 3]. The attempt to automate all types of accounting leads to so-called "universal" corporate information systems [3], which create a single accounting environment and provide access to all necessary data for analysis and decision support. This approach has many disadvantages [2, 4], which can be eliminated by using separate specialized solutions [3, 5]. But this path leads to use of several databases (and perhaps DBMS) that require their synchronization [6]. So, in addition to the main functions of the distributed DBMS: input, storage, processing and sharing data

IT&I-2020 Information Technology and Interactions, December 02–03, 2020, KNU Taras Shevchenko, Kyiv, Ukraine
EMAIL: m.dvoretskiy@gmail.com (M. Dvoretskyi); svetag603@gmail.com (S. Dvoretska); gleb.gorban@gmail.com (H. Horban) ; nezdoliy.yura@gmail.com (Y. Nezdoliy)
ORCID: 0000-0001-5913-6859 (M. Dvoretskyi); 0000-0001-5199-9430 (S. Dvoretska); 0000-0002-6512-3576 (H. Horban); 0000-0002-6003-5585 (Y. Nezdoliy)

– a specific important function is to ensure the collaboration of many users with distributed information [7, 8].

2. Topicality

The database structure optimizing is considered in [9–15], but insufficient attention is paid to improving the automated systems performance by optimizing the structure of the CIS distributed database on the basis of statistics of SQL-queries. Also, in [9–11] while considering the design of automated control and data processing systems, building of data warehouses and multidimensional models, the use of a combined strategy of distributed data representation in CIS is not considered. In [12–15] the authors consider the issue of increasing the productivity of automated systems through the use of materialized views, database restructuring and relations denormalization. However, the optimality of the structure of a single distributed CIS node is ignored. A key factor influencing the reliability and accessibility of the database is the so-called localization of links [5]. If the database is distributed so that the data hosted in a node is called exclusively by its user, it indicates a high level of link localization. If such data distribution is not possible and to execute the user's requests you need to access the information of other nodes, it indicates a low level of links localization.

A combined data distribution strategy is the best in terms of combining the benefits of strategies with and without duplication. But when using it, in addition to the task of synchronizing duplicate information, the task of designing the structure of the database is actual, depending which node data belonging to. In addition, the performance of the system will directly depend on the decision on the need for partial or complete duplication of data. Some tables of a relational database can be duplicated completely, and some – after projection and selection. That is, for optimized data representation on a remote node, it is necessary to use vertical and horizontal data fragmentation procedures.

Therefore, the issue of data distribution between nodes of distributed and territorially dispersed CIS is quite important. Therefore, the task of optimizing the structure of the database of a geographically remote node in corporate information systems is relevant.

3. Purpose of publication

The purpose of the research is to create a mathematical optimization model and subsequent choosing the best alternative to the marker of data representation of the remote node of distributed CIS. The research is related to only to relational databases. The relational data model is based on a simple and at the same time powerful mathematical apparatus, based mainly on theory of sets and mathematical logic [10, 16]. So, when building a mathematical model, it is considered appropriate to use the basic concepts of set theory.

The developed model should take into account the statistics of user requests to local and remote data. Using filtering by selected dimensions, the appropriate subsets of data can be obtained [17]. For dimension elements, the term "data representation marker" was proposed, which determines the level of their need at the node of the distributed corporate information system (DCIS). From the value of this marker, aggregated on the database subset, corresponding to the remote node, will depend on the values of the criteria of model efficiency. It is independence from the central node of the database, the size of the local database and the level of data synchronization [18, 19]. Therefore, one of the tasks is the mathematical representation of the optimality criteria dependence on the value of the data representation marker.

The obtained multicriteria problem must be solved to determine the optimal level of data representation marker. It should be noted that the optimality criteria, the models of which were defined, are independent, monotonic and are represented on the set of real numbers in the interval [0; 1]. The classical methods of Pareto and Slater [20, 21] can give results only at the first stage of modeling. But when calculating the optimal level of the data representation marker they are ineffective due to the decrease in the level of one criteria while increasing others. The solution of the problem is also complicated by the fact that the solution space is defined on a set of real numbers, and therefore the set of solutions contains a large number of alternatives.

4. The main part

Among the well-known relational algebra operations [10], due to the horizontal and vertical fragmentation of data on the distributed CIS node, the operations "projection" (hereinafter P) and "selecting" (hereinafter S) are considered here. Let tup – be a tuple of the relation R, tup[P] be a part of this tuple containing only the values of the attributes that are included in the subset P of the relation scheme R_{shema} ($P \subset R_{shema}$). Then the projection of R on P will be the relation, consisting of tuples of all values from the set P, which exists in the relation R, i.e. $R[P] = \{tup[P] \mid tup \in R_{data}\}$. The scheme of the resulting set can be defined by the following set of attributes: $R[P]_{shema} = \{A_1 \ldots, A_m\}$, where $A_i \in R_{shema}$. The selection displays tuples, and the result is a relation containing a subset of all unique tuples of the relation R, for which a certain logical condition is true $R[S] = \{tup \mid tup \in R_{data} \wedge F(tup, S) = true\}$, where S is a logical condition of SQL-query, and F(tup, S) is a function that reflects its fulfillment for the corresponding tuple. The scheme of the resulting set will equal to the scheme of the basic relation, i.e. $R[S]_{shema} = R_{shema}$. Within the SQL-query for data selecting, a number of relations can be involved, all of which are the result of sequential execution of select and projection operations to the base relation (database table). R" = R'[P], where R' = R[S], i.e.

$$R'' = \{tup[P] \mid tup[P] \in R[P]_{data} \wedge F(tup, S) = true\} \quad (1)$$

Considering the set of queries to the database, the resulting subset $R"_{union}$ of the base relation R can be defined as the union of subsets R' of all queries received by the database from a remote node

$R"_{union} = \bigcup_{i=1}^{n} R_i''$, or

$R"_{union} = \{tup[P_{union}] \mid tup[P_{union}] \in R[P_{union}]_{data} \wedge F(tup, S_{union}) = true\}$,

where $tup[P_{union}] = \bigcup_{i=1}^{n} tup[P_i]$, and $S_{union} = \bigvee_{i=1}^{n} S_i$

To avoid the need for further replication some data that required on the DDB node can only be presented on the central node of the database and participate the query through the use of distributed queries. So the resulting relation R^{remote} will only be a subset of $R"_{union}$. Due to the fact that to represent the data on the remote node it is necessary to use elements of both vertical and horizontal data fragmentation (both projection and selecting), a subset of the base relation R that will describe the relation of the remote node can be represented as follows:

$$R_{shema}^{remote} = \{A \mid A \in R_{shema}, R_{primary} \subset R_{shema}^{remote}, A \in R_{primary} \vee F_a(Node, A) = true\} \quad (2)$$

To make a decision on the attribute representation on a node, the function Fa(Node, A) will be used. Besides, the set of attributes of the relation primary key in any case must be represented on the remote node. The set of tuples, in turn, will be determined by the formula:

$$R_{data}^{remote} = \{ tup \mid tup \in R_{data}, tup_{primary} \in R^{remote\text{-}dep}{}_{data} \vee F_{tup}(Node, tup) = true\} \quad (3)$$

As we can see, the tuple must be represented in the case of entering its primary key to the set of these relations, depending on the current. Otherwise, the need for data is solved using the evaluation function F_{tup}(Node, tup).

The model of presenting user queries should support the possibility of their further classification according to belonging to a particular workplace, location, user role and other criteria that can be added to the model. That is, the user query is defined as

$$Q = <Workplace, User, Application, R_{set''}, Q_{set}{}^{inner}>, \quad (4)$$

where workstation = <Type, Location>; User = <Role, Name>; $R_{set}" = \{ R" \mid \{tup[P] \mid tup[P] \in R[P]_{data} \wedge F(tup, S) = true\} \}$ – the set of resulting relations obtained from the basic relations (tables) of the database by the corresponding queries; Q_{set}^{inner} – is a set of nested queries of the main query Q.

When planning the structure of the database of the remote node of distributed CIS, several factors will be involved - availability and speed of data obtaining, independence from the central DB node, the DB size, the level of data reliability, the need for further synchronization.

In the first step, the simulation begins with the presentation in the remote node the complete copy of the central node DB. In this case, the data availability and independence from the central node of

the database has a maximum level. The speed of data obtaining compared to the central node is usually lower due to less powerful computing resources, but can be increased by performing selecting and projection operations and decreasing the number of data locks. The local database is large, therefore this criteria is not optimal. Also, all data requires synchronization with the central node, which is quite a resource-intensive operation.

The second step is to exclude all unnecessary data from the remote node. To solve this problem, on the basis of a relational model of user SQL-queries (4) was created a multidimensional database [22] with following set of dimensions: <DateTime, WorkplaceType, Location, UserRole, Application, R, A, tup>. For the dimensions elements the term of data representation marker is proposed. It reflects the level of data representation necessity at the node of distributed CIS. For each element value of marker is taken from the following set: {"necessary", neutral", "not required"}. To dimension the "Location", the marking is performed automatically with the value "necessary" for the corresponding remote node and "not required" for all others.

When determining the value of the representation marker for the row of the fact table [22], the max function is used, which reflects the principle of absorption. Determining the value of the marker when performing the consolidation of rows of the fact table on the values of <R, A, tup> (for the table cell) can be performed by moving average method. But the question of the specific influence of each dimension remains unresolved. In addition, it should be taken into attention, that for some subsets of dimensions pessimistic scenario should work (data is needed, no matter what), and for some - optimistic (data should not be duplicated in any case).

So, we have a model where each dimension attribute has a value, a marker and a weight A_{dim} ={Val, Mrk, vol}, where Mrk = {"obligatorily", "necessary", " neutral", "not required" , "forbidden"}, and vol – weight (ignored for the values of the marker "obligatorily" and "forbidden"). By converting a non-numeric linguistic variable of markers into a numeric value ("obligatorily" – "2", "necessary" – "1", "neutral" – "0", "not required" – "-1", "forbidden" – "-2"), the aggregation function was defined:

$$Aggregate_{i=1}^{n} Mrk_i = \begin{cases} 2, \ if \ \exists \ Mrk_i = 2 \\ -2, if \ \exists \ Mrk_i = -2 \ \wedge \nexists \ Mrk_i = 2 \\ \sum_{i=1}^{n} (Mrk_i * \frac{Vol_i}{\sum_{i=1}^{n} Vol_i}) \end{cases} \quad (5)$$

When deciding on the data representation on a remote node, we consolidate the rows of the fact table by the tuple <R, A, tup> and calculate the value of the marker for each of its elements by formula (5). And based on following the decision about data representation is made:

$$Repr\ (Node,R,A,tup) = (Aggregate(R, A, tup)_{i=1}^{n} Mrk_i > koef_{repr}^{node}), \quad (6)$$

where $koef_{repr}^{node}$ – the threshold coefficient of data representation in a certain node Node, that is defined at the range of [-1, 1].

The third step is to completely abandon the local database and place all the data on the central node (or, in some cases, in other nodes) of distributed CIS. In this case, we have the maximization of optimality for criteria of the need for data synchronization. That is because there is no duplication of data. The level of reliability is also maximum, and the size of the local database has a minimum value (no local database). But, at the same time, the availability of data and the access speed are minimized, and the work of CIS is highly dependent on the central node availability.

The value of some criteria improved compared to the second step, but at the same time the value of the others got worse. It is logical to assume that the optimal values of all DCIS DB structure criteria acquire between the 2nd and 3rd steps. To be able to perform the analysis and find the optimal distribution of data between the remote and central nodes, it is necessary to formalize the database structure quality criteria.

Criterion of independence from the central database node, and, accordingly, the availability and access speed directly depend on the representation of user SQL-query data on the node of distributed CIS. Using the model of the user SQL-query (4) and the resulting relation of the remote node (1, 2), we can determine the function of the request data availability:

$$F_{availab}(Node,\ Q) = \begin{cases} 1, if\ \forall R'' \exists R''_{shema}, R'' \in R^{remote}_{shema} \wedge \\ \quad \forall Q^{inner} F_{availab}(Q^{inner}) = 1 \\ 0,\ if\ \exists R'' \nexists R^{remote}_{shema}, R'' \in R^{remote}_{shema} \vee \\ \quad \exists Q^{inner} F_{availab}(Q^{inner}) = 0 \end{cases} \quad (7)$$

The aggregate value of the data availability level and independence from the central DB is defined as the average value

$$F_{availab} = \frac{\sum_{i=1}^{n} F_{availab}(Q_n)}{n},\ where\ Q_n \in Q_{node}. \quad (8)$$

The set of user SQL-queries Q_{node} is a subset of all user queries Q_{all} ($Q_{node} \subset Q_{all}$), where for each element the function of belonging to a remote node is equal to one.

$$Q_{node} = \{Q \mid F_{availab}(Node, Q) = 1\},$$

where

$$F_{availab}(Node, Q) = \begin{cases} 1, if\ (\exists R'' \in R''_{set} \\ \rightarrow Aggregate(R'')_{i=1}^{n} Mrk_i > -1) \vee \\ \quad (\exists Q^{inner} \in Q^{inner}_{set} \\ \quad \rightarrow F_{availab}(Node, Q^{inner}) = 1) \\ \\ 0, if\ (\forall R'' \in R''_{set} \\ \rightarrow Aggregate(R'')_{i=1}^{n} Mrk_i \leq -1) \wedge \\ \quad (\forall Q^{inner} \in Q^{inner}_{set} \\ \quad \rightarrow F_{availab}(Node, Q^{inner}) = 0) \end{cases}$$

Next, we consider the criterion of the local database size. This criterion affects both the performance of queries to the local database and the power of computing resources required to perform database and CIS administration operations. The database under the relational DBMS control (including distributed) is presented on disk space as a file or group of files [7, 8]. At the same time, any modern relational DBMS has mechanisms for obtaining information about how much disk space is used by each relation. In the vast majority of cases, the total value of the relations size equals the total value of the database files sizes.

But the information about size of R does not make it possible to determine the size of R", which is the result of a sequence of selecting and projection operations, and is part of the set R^{remote}. On the other hand, each DBMS provides information about the amount of disk space required to store the value of the attribute defined on a particular domain [7, 8]. The size of the tuple can be determined as

$$Size_R = SizeR0_i^{DBMS} + p\ x \sum_{i=1}^{n} Size(Type_i), \quad (9)$$

where $A_i \in D_i \in Type_i$, p – is the relation power, and $SizeR0_i^{DBMS}$ – is the size of the i-th relation if it is empty.

However, the values obtained by (9) cannot be used in calculations, because $Size_R$ almost never equals to $SizeR^{dbms}$. This may be due to the presence of additional data structures (indexes) related to the table, as well as other properties of data representation on the disk. Therefore, for each relation we determine the correction factor

$$Koef_{sizeR} = \frac{SizeR_i^{DBMS} - SizeR0_i^{DBMS}}{p \times \sum_{i=1}^{n} Size(Type_i)} \quad (10)$$

Next, when determining the size of R " (subset of R) we use the following formula

$$Size_{R''} = Koef_{sizeR} \times p' \times \sum_{i=1}^{n'} Size(Type_i) \quad (11)$$

where p is the power of R'', n' – is the number of elements of the set $R^{remote}{}_{shema}$ (number of attributes), and each attribute $A_i \in D_i \in Type_i$.

But for each individual case of the subject area, the size (11) will take different values, and therefore its absolute value has no sense. Therefore, it was decided to present the final value of the criterion of the local database size in proportion to the size of the database in the CIS central node.

$$F_{size} = \sum_{i=1}^{n} \frac{Size_{R'',i}}{SizeR_i^{DBMS}} \quad (12)$$

The last of the above criteria is the need for data synchronization. First, we define a subset of the remote node data for which the data change operations are performed. To do this, define the model of the SQL-query that modify data Q^{modif} = <Виміри, R''^{modif}, type>, where R''^{modif} – is a subset on the relation R, which changes due to data modification operations, type = {insert, update, delete} – operation type. R''^{modif} is defined as

$$R''^{modif} = \{tup[P^{modif}] / tup[P^{modif}] \in R[P^{modif}]_{data} \wedge F(tup, S) = true\} \quad (13)$$

where S – is a logical condition, defined in SQL query, F(tup, S) – is a function that reflects its fulfillment for the corresponding tuple, and P^{modif} – is a set of attributes that are modified.

Considering the set of queries to the database, the resulting subset R''^{modif}_{node} of the base relation R can be defined as the union of subsets R''^{modif} of all queries (13) received by the database from the remote node $R''^{modif}_{node} = \bigcup_{i=1}^{n} R_i''^{modif}$.

Similarly, we define the set R''^{modif}_{main}, which will be modified on the central node or other nodes with future synchronization with the central node. The intersection of the sets R''^{modif}_{node} та R''^{modif}_{main} will determine the subset of the basic relation on which data conflitct can take place. This data require the use of more resource-intensive synchronization algorithms [6].

$$R'''^{modif}_{node} = R''^{modif}_{node} \cap R''^{modif}_{main} \quad (14)$$

Based on (14), we add to the multidimensional DB (5) the dimension SyncroFlg = {true, false}, which will be determined on the tuple <R, A, tup>. Next, based on the aggregate value of the representation marker $Aggregate_{i=1}^{n} Mrk_i$ and the representation coefficient $koef_{repr}^{node}$ perform filtering of the multidimensional DB according to the decision on representation (6) and SyncroFlg = true. Aggregate the results by <R, A, tup> and count the number of queries. The ratio of the obtained value to the total number of queries according to (6) will be an indicator of the level of data synchronization need

$$F_{synchro} = \frac{p_{node}^{modif}}{p_{node}}, \quad (15)$$

where p_{node}^{modif} – relation power, including queries of the remote node (according to the decision on representation), which includes the values of the tuples attributes (cells), which are also included in the set R''^{modif}_{node}, and p_{node} – the cardinality of all queries, attributes and tuples of which are represented in the remote node.

A multicriteria problem, that was obtained, must be solved to determine the optimal level of data representation marker. Classical Pareto and Slater methods [20, 21] can give results only at the first stage. But when calculating the optimal level of data representation marker are ineffective due to the decrease in the level of some criteria of optimality while increasing others. The solution of the problem is also complicated by the fact that the solution space is determined on a set of real numbers, and therefore the set of solutions contains many alternatives. The analytic hierarchy process (AHP), which is a general methodology for solving a wide class of decision-making problems, allows to combine a relatively simple mathematical apparatus with knowledge and experience of the decision maker. The basis of this method is the representation of the decision process in the form of a multilevel hierarchy. This hierarchy should reflect all the components of the problem to be solved. The method is based on the principles of decomposition, pairwise comparisons and hierarchical

composition. The main stages of the method are building a hierarchy, estimating the importance and priorities, checking the consistency of priorities and synthesis of the solution.

When compiling the hierarchy, following relationship between the levels elements was used: goal - stakeholders - criteria - alternatives. The value of the data representation marker (alternative) is a real number in the interval [-1, 1]. It leads to potential large number of alternatives at the 4th level of the hierarchy and therefore the matrices of pairwise comparisons by criteria can become very big. This complicates estimation process for the decision makers. It is proposed to simplify the task by reducing the number of alternatives to 5: "low" (L) – "-1", "lower them medium" (LM) – "-0.5", "medium" (M) – "0", "higher then medium" (HM) – "0.5", and "high" (H) – "1". The level of "decision makers" is represented by the elements "Owner", "Database Administrator", "Database Developer" and "CIS Operator". The obtained hierarchical model is shown in Fig. 1.

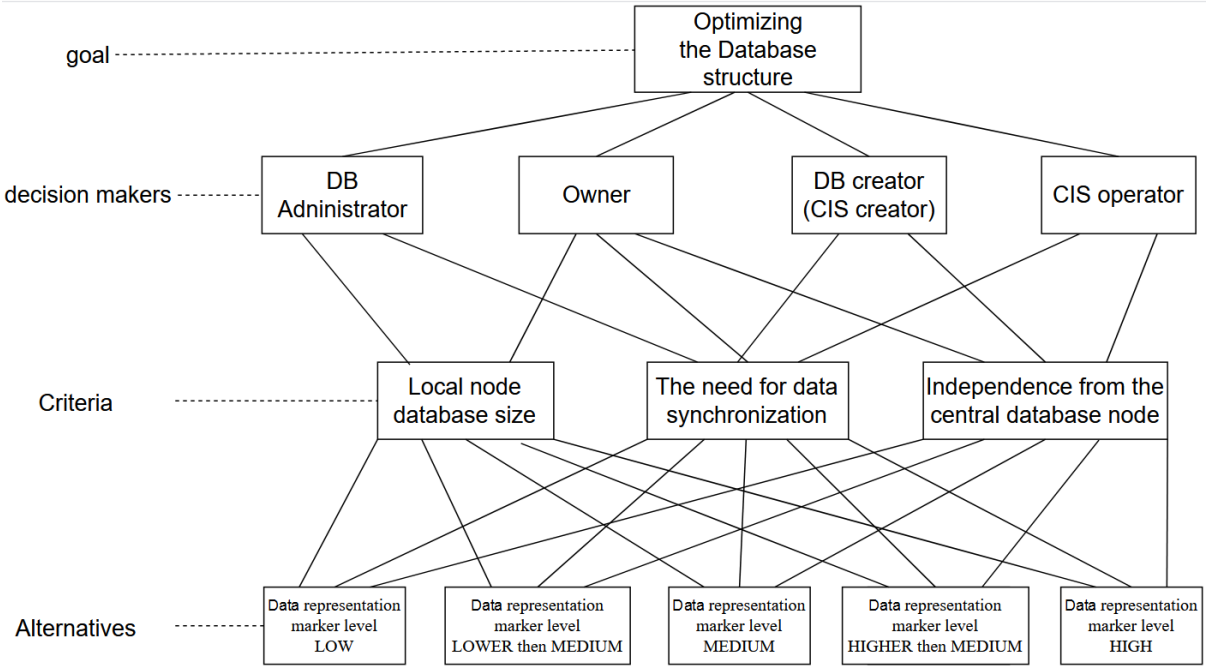

Figure 1: Hierarchical model of the distributed CIS node structure optimization problem

Note that the list of criteria differs for the decision makers. Thus, all three criteria are important for the owner (the database size, the need for synchronization and independence from the central database), because they have influence on both the quality of CIS and the cost of equipment. For the database administrator, the criteria of database size and the need to organize data synchronization are important. In turn, for the database developer and CIS operator, the criterion of database size is not critical. It is clear that the relative weight of each of the criteria for different decision makers will also differ. Using the scale of relative importance of the criteria [23] and with the involvement of the decision maker (which at this stage is the owner) we build a matrix of pairwise comparisons for decision makers (Table 1). At the third level of the hierarchy, the corresponding matrices of pairwise comparisons are formed according to the criteria of optimality for each decision maker. Thus, for the decision maker "owner" we have the following matrix of pairwise comparisons of optimality criteria (Table 2).

Table 1

The matrix of pairwise comparisons for decision makers

	Owner	DB Admin	DB developer	CIS operator
Owner	1	3	5	7
DB Admin	1/3	1	3	5
DB developer	1/5	1/3	1	3
CIS operator	1/7	1/5	1/3	1

Table 2

The matrix of pairwise comparisons of optimality criteria for the decision maker "owner"

	BD size	Independence level	Need for synchronization
BD size	1	1/7	1/3
Independence level	7	1	5
Need for synchronization	3	1/5	1

To check the conflicts existence between matrix elements, the consistency index (CI) is calculated. For the data in Table 2 CI = 3.2%, which indicates the allowable level of consistency (in case the value is higher 10% there is a need to adjust the values of the matrix).

The next step in the classical analytic hierarchy process is to fill in the matrices of pairwise comparisons of alternatives separately for each criterion of optimality, similar to Table 1 and Table 2. In our case, the presence of mathematical models for calculating the values of the optimality criteria formulated in (8, 12, 15) allows to perform the initial calculation of matrix data based on numerical values of the data representation marker for each alternative. Next, the matrix is submitted to the decision maker for approval. For example, the size of the local node database depending on one of the five alternatives can change as follows (Table 3).

Table 3

Dependence of the database size on the selected alternative

Marker level	Low	Lower then medium	Medium	Higher then medium	High
DB size	0,02	0,24	0,47	0,55	0,75

Based on the above data, the size of the database at low (min) and high (max) level of the data representation marker differs by 0.75 / 0.02 = 37.5 times. According to principles of pairwise comparisons and the axiom of homogeneity, we perform normalization of the values given in Table 3, using a slightly modified formula of natural normalization:

$$W_i^{norm} = \frac{(W_i - min_i W_i)}{(max_i W_i - min_i W_i)} * (k-1) + 1, \quad (17)$$

where W_i is the value of the optimality criterion for the i-th alternative, and k = 9.

Normalized according to (17) the database size values (Table 3) are presented in Table 4.

Table 4

Normalized criteria values of the database size

Marker level	Low	Lower then medium	Medium	Higher then medium	High
DB size (Normalized)	1	3,41	5,93	6,81	9

After rounding to the integer according to mathematical rules, we build a matrix of pairwise comparisons of alternatives for the criterion of the local database size (Table 5).

According to (4) we perform the calculation of the matrix of alternatives relative weight by the criterion of the local database size. Also, we similarly calculate the priority vectors of alternatives according to the criteria of independence from the central node and the need for data synchronization. As a result, we obtain following vectors.

$$W_{owner}^{sizeDB} = \begin{bmatrix} 0{,}570 \\ 0{,}190 \\ 0{,}095 \\ 0{,}081 \\ 0{,}063 \end{bmatrix}, W_{owner}^{Independ} = \begin{bmatrix} 0{,}036 \\ 0{,}082 \\ 0{,}164 \\ 0{,}328 \\ 0{,}328 \end{bmatrix}, W_{owner}^{Synchro} = \begin{bmatrix} 0{,}082 \\ 0{,}063 \\ 0{,}101 \\ 0{,}184 \\ 0{,}550 \end{bmatrix} \quad (18)$$

Table 5
Dependence of the database size on the selected alternative

DB size	Low	Lower then medium	Medium	Higher then medium	High
Low	1	3	6	7	9
Lower then medium	0,33333	1	2	2	3
Medium	0,16667	0,5	1	1	2
Higher then medium	0,14286	0,5	1	1	1
High	0,11111	0,33333	0,5	1	1

According to (16) and (18) we calculate the global vector of priorities for the decision maker "owner" (19).

$$W_{owner}^{glob} = \begin{bmatrix} 0{,}09 \\ 0{,}09 \\ 0{,}15 \\ 0{,}30 \\ 0{,}37 \end{bmatrix} \tag{19}$$

By performing the appropriate calculations, we obtain global priority vectors for other decision makers

$$W_{DB\ admin}^{glob} = \begin{bmatrix} 0{,}41 \\ 0{,}15 \\ 0{,}10 \\ 0{,}12 \\ 0{,}23 \end{bmatrix},\ W_{DB\ dev}^{glob} = \begin{bmatrix} 0{,}04 \\ 0{,}08 \\ 0{,}17 \\ 0{,}33 \\ 0{,}38 \end{bmatrix},\ W_{CIS\ oper}^{glob} = \begin{bmatrix} 0{,}05 \\ 0{,}08 \\ 0{,}16 \\ 0{,}31 \\ 0{,}40 \end{bmatrix} \tag{20}$$

Using the obtained results of the global priorities vectors for decision makers (19), (20) and the matrix of preferences of decision makers (Table 1), we calculate the vector of global priorities of alternatives (Table 6).

Table 6
Calculation of global priorities of alternatives

	Owner	CIS operator	DB Admin	DB developer	Global priorities
	0,563	0,055	0,263	0,117	
Low	0,09	0,05	0,41	0,04	0,17
Lower then medium	0,09	0,08	0,15	0,08	0,1
Medium	0,15	0,16	0,1	0,17	0,14
Higher then medium	0,3	0,31	0,12	0,33	0,26
High	0,37	0,4	0,23	0,38	0,34

The performed calculations allow to organize decision support when choosing the optimal level of the data representation marker among the proposed alternatives.

5. Summary and conclusion

Based on the relational data model, the concept of data slices of the set of database relations is formalized. Using the definition of selecting and projection operations, as well as taking into account the hierarchical structure of user queries, the model that describes their structure was built. This model includes analytical characteristics and allows to define for each base relation a subset (node relation), which will consist of elements that are part of the resulting sets of SQL-queries sequence.

The term of data representation marker for elements of analytical dimensions was proposed. Using the offered aggregation function the level of representation marker for each attribute and tuple of relation is calculated. To determine the optimal value of the representation marker, several optimality criteria are introduced and mathematical models are built for each of them. This allow to calculate their values depending on the limit level of the data representation marker at the node of distributed CIS. Solving a multi-criteria problem and finding the optimal level of data representation at a remote node can increase the level of data availability and efficiency of distributed CIS. Efficiency is defined as the ratio of result and resources, so taking into account the vector of relative weight of the optimality criteria of the model (16), we calculate the efficiency as

$$Eff = \frac{F_{availab} \times W_1^{criteria}}{F_{size} \times W_0^{criteria} + F_{synchro} \times W_2^{criteria}} \quad (15)$$

The comparison of the obtained results for the database of the KIS node of the subject area is given in Table 7.

Table 7

Comparison of the database structure effectiveness in different strategies and levels of data representation at the node of the distributed CIS

		Using central node DB	Presenting only critical data	Presenting of all necessary data	Full data duplication	Optimal level of data representation marker
Independence	0,73	0	0,35	0,97	1	0,97
DB size	0,081	0	0,02	0,75	1	0,63
Synchro need	0,188	0	0,15	0,07	1	0,08
DB node efficiency		-	8,5589	9,5892	2,7126	10,7257
Efficiency increase,%		-	**25,32%**	**11,85%**	295,41%	-

Thus, the results of the research allow to increase the efficiency of using the distributed CIS node of the subject area by 25% compared to the presentation of only critical data, and by 11% compared to the presentation of all necessary data of the central database, respectively. The research can be followed by presenting the obtained vector of global priorities in the form of fuzzy sets of one variable. Dephasing the obtained results can make numerical value of the optimal level of data representation at the RKIS node more accurate.

6. Acknowledgements

This research was partially supported by the state research projects: “Development of information and communication decision support technologies for strategic decision-making with multiple criteria and uncertainty for military-civilian use” (research project no. 0117U007144, financed by the Government of Ukraine); “Development of information-analytical system for military-civil application as a information protection factor in the conditions of multi-criteria, uncertainty and risk” (research project no. 0120U101222, financed by the Government of Ukraine).

7. References

[1] M. Tamer Özsu, Patrick Valduriez. Principles of Distributed Database Systems 3rd ed. Springler, 2011.

[2] M. Dvoretskyi, S. Borovlova, Web-application of warehouse accounting in non-automated points of sale, Science works "Petro Mohyla Black Sea National University", Rel. 308. T. 320, Series: Computer technologies, 2018, pp. 45–52 (in Ukrainian).

[3] 1C: Enterprise 8. Management of a trade enterprise for Ukraine, [Online]. Availably: http://rarus.com.ua/torgovyy-i-skladskoy-uchet/1S_Predpriyatie_8_Upravlenie_torgovym_predpriyatiem_dlya_Ukrainy_/ (in Russian)
[4] Natalia Kozliuk, Svetlana Uhrymova, Warehouse accounting at trade enterprises, Phenix, 2005. (in Russian)
[5] H. Garcia-Molina, J. D. Ullman, and J. Widom, Database Systems: The Complete Book 2nd Edition, Pearson, 2008.
[6] Automatic synchronization of distributed databases in split mode, [Online]. Availably: http://stimul.kiev.ua/materialy.htm?a=avtomaticheskaya_sinkhronizatsiya_raspredelennykh_baz_dannykh_v_razdelennom_rezh (in Russian)
[7] Maksym Kuznetsov, Yhor Symdianov, MySQL 5, BHV-Piterburg, 2010. (in Russian)
[8] Dusan Petkovich. Microsoft SQL Server 2019: A Beginner's Guide, Seventh Edition 7th Edition, Kindle Edition, Mc-Graw-Hill Education, 2020.
[9] V. V. Pasichnyk, V. A. Reznichenko, Organization of databases and knowledge bases, Publishing group BHV, 2006. (in Ukrainian)
[10] Ye. V. Malakhov, Fundamentals of database design: Textbook for students of higher education institutions, Science and technology, Odesa, 2006. (in Ukrainian)
[11] V. V. Pasichnyk, N. B. Shakhovska, Data warehouses: a textbook, Magnoliya, Lviv, 2008. (in Ukrainian)
[12] A. B. Kunhurtsev, Yu. N. Vozovykov, Finding Patterns in Query Distribution to Manage Materialized Views, in: Proceedings of the Odessa Polytechnic University, Publishing house Odessa National Polytechnic University, Odesa, 2(30), 2008, pp. 135–140. (in Russian)
[13] A. B. Kunhurtsev, S. L. Zynovatnaia, Relational database restructuring model by denormalizing the schema of relations, in: Proceedings of the Odessa Polytechnic University, Publishing house Odessa National Polytechnic University, Odessa, 2(26), 2006, pp. 105–111. (in Russian)
[14] V. A. Filatov, R. V. Semenets, Methods and tools for designing information systems and distributed databases, Bulletin of Kherson National Technical University, Kherson, No 4(27), 2007, pp. 203–207. (in Russian)
[15] S. V. Lazdyn, S. Yu. Zemlianskaia, Optimization of distributed corporate information networks using genetic algorithms and object modeling, Scientific works DonNTU, No 147, 2009, pp. 83–95. (in Ukrainian)
[16] Rebecca M. Riordan. Designing Relational Database Systems (Dv-Mps Designing), Microsoft Press, 2001.
[17] M. Dvoretskyi, S. Dvoretska, Information technology for determining useful data while optimizing the structure and minimizing the volume of the distributed database node, Bulletin of Cherkasy State Technological University, Cherkasy, 4/2019, 2019, pp.26–35. (in Ukrainian)
[18] M. Dvoretskyi, S. Dvoretska, Y. Nezdoliy, S. Borovlova, Data Utility Assessment while Optimizing the Structure and Minimizing the Volume of a Distributed Database Node, in: Proceedings of the 1st International Workshop on Information-Communication Technologies & Embedded Systems (ICTES 2019), Mykolaiv, 2019. pp. 128–137
[19] M. Fisun, M. Dvoretskyi, A. Shved and Y. Davydenko, Query parsing in order to optimize distributed DB structure, in: Proceedings of the 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Bucharest, 2017, pp. 172–178. doi: 10.1109/IDAACS.2017.8095071
[20] I. Kovalenko, Y. Davydenko, A. Shved, Searching for Pareto-Optimal Solutions. In: Shakhovska N., Medykovskyy M. (eds) Advances in Intelligent Systems and Computing IV. CCSIT 2019. Advances in Intelligent Systems and Computing, vol 1080. Springer, Cham, 2020, pp. 121–138. doi: 10.1007/978-3-030-33695-0_10
[21] A. Shved, I. Kovalenko, Y. Davydenko, Method of Detection the Consistent Subgroups of Expert Assessments in a Group Based on Measures of Dissimilarity in Evidence Theory. In: Shakhovska N., Medykovskyy M. (eds) Advances in Intelligent Systems and Computing IV. CCSIT 2019. Advances in Intelligent Systems and Computing, vol 1080. Springer, Cham, 2020, pp. 36–53. doi: 10.1007/978-3-030-33695-0_4
[22] A. Barsegyan, M. S. Kupriyanov, V. V. Stepanenko, I. I. Holod, Methods and models of data analysis: OLAP and Data Mining, BHV-Petersburg, 2004. (in Russian)