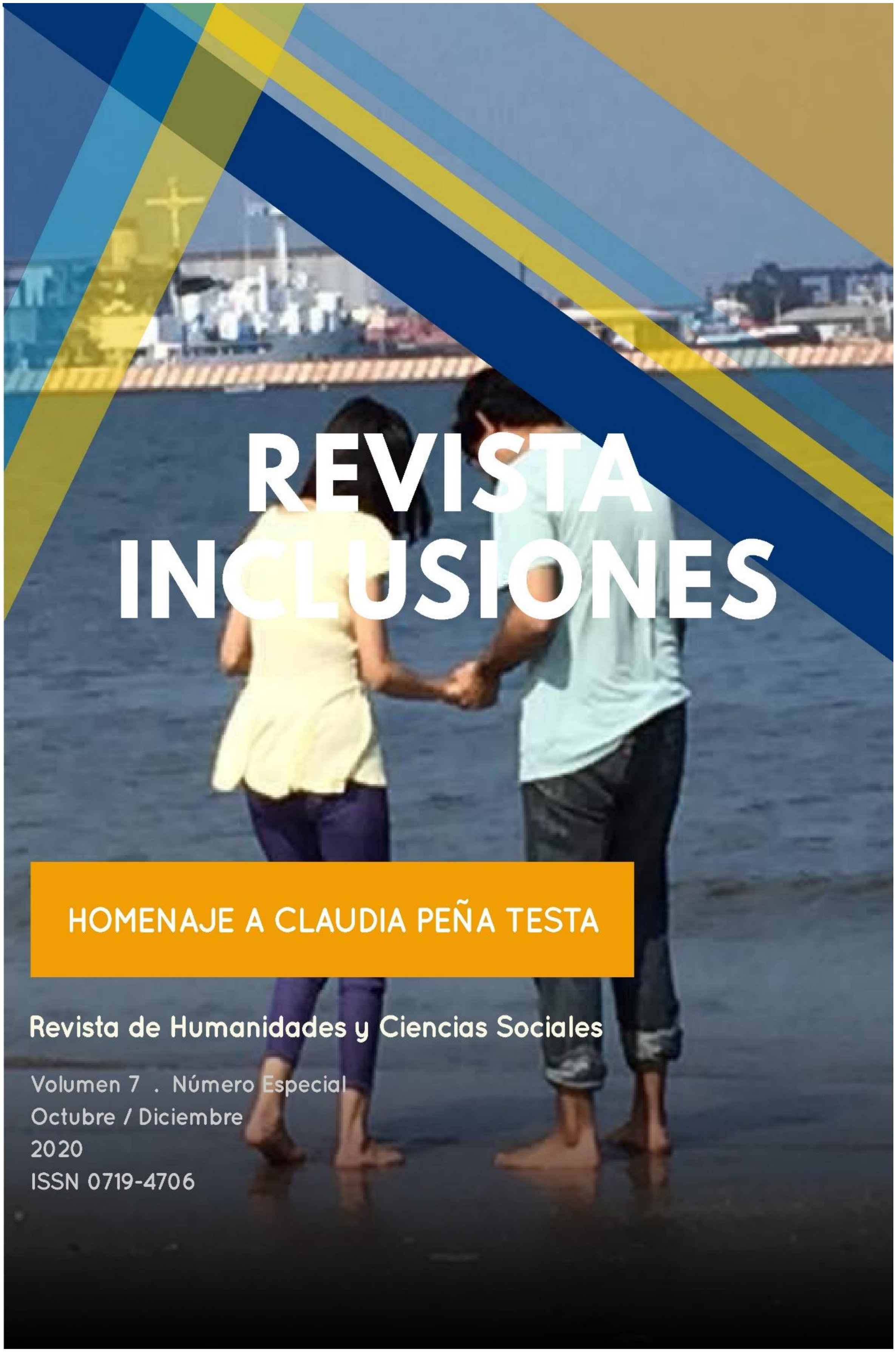


REVISTA INCLUSIONES

HOMENAJE A CLAUDIA PEÑA TESTA

Revista de Humanidades y Ciencias Sociales

Volumen 7 . Número Especial
Octubre / Diciembre
2020
ISSN 0719-4706

REVISTA INCLUSIONES M.R.
REVISTA DE HUMANIDADES
Y CIENCIAS SOCIALES

CUADERNOS DE SOFÍA
EDITORIAL

REVISTA
INCLUSIONES M.R.
REVISTA DE HUMANIDADES
Y CIENCIAS SOCIALES

CUADERNOS DE SOFÍA
EDITORIAL

REVISTA
INCLUSIONES M.R.
REVISTA DE HUMANIDADES
Y CIENCIAS SOCIALES

CUADERNOS DE SOFÍA
EDITORIAL

Editorial Cuadernos de Sofía
Santiago – Chile
OBU – C HILE

Indización, Repositorios y Bases de Datos Académicas

Revista Inclusiones, se encuentra indizada en:

Web of Science Group

Clarivate Analytics Emerging Sources Citation Index Indexed

Clarivate Analytics

DOAJ Directory of Open Access Journals

MIAR 2014 Live

latindex 15 años

CATÁLOGO

NSD ERIH PLUS European Reference Index for the Humanities and Social Sciences

Plataforma Sucupira

CAPES QUALIS

Indexed by LatAm Estudios

Actualidad Iberoamericana

biblat Bibliografía Latinoamericana en revistas de investigación científica y social

CLASE Citas Latinoamericanas en Ciencias Sociales y Humanidades

CiteFactor Academic Scientific Journals

Academic Resource Index ResearchBib

WorldCat®

SHERPA/RoMEO

ROAD Directory of Open Access Scholarly Resources

Princeton University Library

ubl

Stanford University Libraries

Uniwersytet Wrocławski

uOttawa Bibliothèque Library

Paris biu santé

WZB Berlin Social Science Center

TÜBİTAK ULAKBİM

VPL Vancouver Public Library

UNB Est. 1785 Libraries

Jülich Forschungszentrum

IPN

MLZ
Heinz Maier-Leibnitz Zentrum

REX

UNIVERSITY OF SASKATCHEWAN

I2OR

en Diálogos Mercosur

SIS

International Innovative Journal Impact Factor (IIJIF)

DRJI Directory of Research Journals Indexing

WESTERN
THEOLOGICAL SEMINARY

BIBLIOTECA ELECTRÓNICA DE CIENCIA Y TECNOLOGÍA

Universidad de Concepción

BIBLIOTECA UNIVERSIDAD DE CONCEPCIÓN

Hellenic Academic Libraries Link
HEAL LINK
Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών

REVISTA INCLUSIONES M.R.
REVISTA DE HUMANIDADES
Y CIENCIAS SOCIALES

CUADERNOS DE SOFÍA
EDITORIAL

ISSN 0719-4706 - Volumen 7 / Número Especial / Octubre – Diciembre 2020 pp. 189-207

OPTIMIZATION OF DATA WAREHOUSE MODEL ON THE BASIS OF ATTRIBUTE CLUSTERING

Lic. I. V. Belchenko
Armavir State Pedagogical University, Russia
ORCID: 0000-0001-6289-7843
ilur@mail.ru
Dr. R. A. Dyachenko
Armavir State Pedagogical University, Russia
ORCID: 0000-0003-1244-1228
emessage@rambler.ru
Ph. D. (c) V. E. Belchenko
Armavir State Pedagogical University, Russia
ORCID: 0000-0003-0955-1370
bvee@mail.ru
Dr. V. A. Atroshchenko
Armavir State Pedagogical University, Russia
ORCID: 0000-0003-4575-9048
atros.v@yandex.ru
Ph. D. (c) E. P. Chernyaeva
Armavir State Pedagogical University, Russia
ORCID: 0000-0001-7745-7090
ellacher1982@bk.ru

Fecha de Recepción: 17 de junio de 2020 – **Fecha Revisión**: 27 de junio de 2020

Fecha de Aceptación: 28 de septiembre 2020 – **Fecha de Publicación**: 01 de octubre de 2020

Abstract

Performance of information system is one of the main indices of its efficiency. Most existing information systems and software complexes operate with relational data warehouses. Design of data warehouses for information system is comprised of sequential development of conceptual, logical, and physical models. This design sequence has proved its efficiency in the case when the requirements to database, defined by domain area, are strictly formalized, the load on data processing center is forecastable. However, there are cases when several information systems operate with data warehouse. Each of them has its own requirements to information, sometimes contradictory, which should be presented by database in the shortest possible time. In such cases, forecasting of load, detection of the most resource consuming requests require for system analysis.

Keywords

Decision support system – Optimization – Data structures – Data warehouse – System analysis

Para Citar este Artículo:

Belchenko, I. V.; Dyachenko, R. A.; Belchenko, V. E.; Atroshchenko, V. A. y Chernyaeva, E. P. Optimization of data warehouse model on the basis of attribute clustering. Revista Inclusiones Vol: 7 num Especial (2020): 180-207.

Introduction

Online communication with user, minimization of system response time are the requirements to most information systems based on data warehouses. Execution of the requirements depends not only on hardware comprised of servers and communication lines, but also on software components including web applications and data warehouse. The most time consuming are disk operations related with position of reading head as well as with data reading time related with interruptions of these operations. Time consumption for these operations in its turn depends on data model, methods of its physical allocation on hard disks.

Relational data warehouse is a combination of data and respective interrelations. Data are arranged in the form of table structures comprised of attributes and sets of attribute values: strings. The table structures contain information about objects of domain area presented in data warehouse. Each column of table structure contains data of certain type corresponding to that of column attribute. Intersection of column and string contains the attribute value. A string of table structure contains a set of interrelated values characterizing the object of domain area. Primary key is the value or set of values of attributes uniquely characterizing the string of table structure. Links between table structures are implemented by linking potential and external keys. These data can be accessed by various requests, table rearrangement is not required. One of the main principles upon determination of structure of physical record is saving of logical record content. Physical record is a combination of linked data corresponding to one or several logical records. Physical record is comprised of two parts: service part and information one. The service part of physical record is used by data warehouse for identification of record, definition of its type, storing deletion identifier, storing tags of record elements, identifier of record length, establishment of structural associations between records, coding element values. User software does not have access to service part of physical record. The fields of information parts contain values of data elements. There are several methods to allocate data in physical record.

Formats of physical records:

- Positional arrangement of physical records. In each record the sequence of elements is the same, the element type is not shown, only its value is stored. The fields have fixed length. The element value in each record copy appears from one and the same position determined in description of file structure of data warehouse. All records have the same length. Upon such record format, it is possible to apply efficient search algorithms (binary method, golden section method, etc.), however, the memory is used inefficiently. Another advantage of the positional arrangement of physical records is the absence of time consumption for determination of end address of one record and start of another one. If the data have less length than the size of record field, then they are justified leftwards or rightwards, free spaces are filled with spaces. The field size is determined at the stage of structure design of data warehouse according to possible values of attributes of domain area. The number of free positions can be reduced by separators or index method.

- Record storage with separator. Record has fields of variable length, hence, the stored record has variable length. According to storing method, it is required to select certain separating symbol which is not used anywhere upon arrangement of database files. For instance, the symbol #. Upon such arrangement of stored records, the memory is used sparingly, however, rapid algorithms of data search cannot be applied. There is a gain in memory but loss in performance.

– Index arrangement of physical records. Upon such method, the start of record elements is determined by index array located in service part of the record. Similar to storage with separator, the index method consumes memory more sparingly, though, does not allow to use efficient data search algorithms. Necessity to read service part in addition to information part, as well as its processing exert negative effect on execution rate of requests for data reading.

– Storage with handles. This method is efficient when record contains many elements, the values of which are absent, and it is necessary to store only elements with known values.

The three latter methods use memory more sparingly, especially when some fields are not filled, though, the search for required information consumes more time.

Record clustering: combination of records of various types in physical groups, which allows to efficiently use the advantage of sequential data allocation. Clustering is allocation of linked data near each other aiming at increase of access efficiency. Upon logical clustering, certain types of data elements are combined into records. Upon physical clustering, the copies of records of the same or different types are combined and allocated into one block, one domain, or onto one storing device, that is, addressed memory of fixed size. The problem of clustering is to determine how to store records of data warehouse for preset model so that to minimize access time for typical operation loads; that is, the data most frequently used in search operations and data manipulation are combined and addressed firstly. In general case, optimum clustering is difficult for complex integrated data warehouses where data addressed by various applications are overlapped and searches by sequential and random methods can be easily compromised. An important notion upon consideration of physical arrangement of data warehouses is the notion of block.

Block is the minimum addressed element of external memory, by means of which data exchange between RAM and external memory is performed. The block-based principle of data presentation is the main for operation system and is characterized by numerous advantages, the most important is fixed size of data transferred by data buses. RAM and ROM data are exchanged in blocks. Writing and reading of blocks are performed via RAM buffer. In order to organize each file of data warehouse, depending on its size in external memory, from one to N blocks are allocated for records. In accordance with the block principle of data storage, the following allocations are possible:

- all records can be placed in one block;
- one record is in several blocks;
- otherwise, one record is in one block.

The time of reading and recording of file elements depends on the selected allocation variant. Each byte in block has its number. The number of block byte, from which the record starts, defines relative record address in the block, that is, the record address equals to the block number plus relative address in the block (consecutive access), or the block number and the key value (here: combination of consecutive and direct access). Records in blocks are located densely, without intervals, consecutively one after another. In a block, part of memory is allocated for service information: relative address of free memory segments, pointers to next block, etc.

For the delivered data, which should be located in one block already filled completely, additional memory block is allocated in record overflow area arranged in the form of one block, where records are linked by pointers in one chain. The overflow area is generally restricted, since the time of data search in the overflow area increases significantly. Thus, as the limit of overflow area is reached, the data warehouse file is rearranged. The file is supplemented by the required number of blocks in main memory and records are repacked. Herewith, the main aim is to free overflow area and to locate all records in partially free blocks (empty or reserved for possible addition of records)[1].

Repacking of physical data blocks is a resource consuming operation, which is followed by rearrangement of indices. Therefore, selection of attribute type, amount of disk space occupied by element is an important multicriterial problem, which is often omitted by designers of data warehouses, thus, the data types with unreasonably high upper limit are selected[2].

For each file the blocks are numbered, and the system determines the required file by file name and block number. Therefore, the search rate depends on block size in bytes, file size, number of records in file block, number of records in index block, number of blocks in file, fraction of reserved part of block, number of fields in record, record size in bytes, length of key field in record.

In accordance with the aim, i.e. decreasing time for requests for data reading from relational data warehouse, the procedures consider positional arrangement of stored records, which provides high search rate due to fixed length of physical records stored in blocks.

The proposed procedures are aimed at optimization of table allocation in relational data warehouse and are comprised of the following initial data and elements of basic sets.

Data types of information system. Data type of certain object is such property of these data, which determines the values to be stored in this object and operations to be executed with these values.

In modern data warehouses the following data types are used:

- symbol (text);
- numerical;
- logical;
- time;
- specialized.

[1] E. I. Chigarina, Databases: guidebook (Samara: SGAU, 2015).

[2] I. V. Belchenko y R. A. Dyachenko, “Metodika povysheniya proizvoditel'nosti informatsionnoi sistemy za schet optimal'noi restrukturizatsii dannykh”, Izvestiya vysshikh uchebnykh zavedenii. Povolzhskii region. Tekhnicheskie nauki num 1 (2018): 26-38.

Selection of data types for information system depends on the expert's opinion in the field interrelated with information system. String types of data with fixed and variable length affect the rate of request execution. Indexation by string fields is faster if they have fixed length. The data files of information systems include various subtypes, which differ in dimensionality but retain all properties of parent type. For instance, data type is string. Subtypes of string type are nchar (10), varchar (255), etc.

Attributes of information system data model. Attribute presents property describing the essence. Attributes correspond to data types of information systems. All data stored in attribute should be of the same type and have same properties.

Memory size occupied by attribute. Each data subtype is characterized by memory size required for copy storage. The issue of selection of optimum attribute subtype is related with response speed of information system and respective relational data warehouse.

D. Gorokhov and V. Chernov in their article titled Methods of data storage in DBMS describe principles of physical data storage in DBMS. The main units of physical storage are data block, extent, file (or hard disk partition). Logical level of data presentation includes spaces (or table spaces). Data block or page is a unit of exchange with external memory. Page size is fixed for data warehouse or for its various structures and is set upon creation. It is very important to select correct block size at once: it is nearly impossible to change it in operating base. The block size exerts high effect on performance of data warehouse: at large sizes the rate of reading/writing increases, especially this is peculiar for complete review of tables and operations of intensive data load, however, charges for storage increase and efficiency of index review decreases. Smaller block size allows more efficient memory consumption; however, it is relatively expensive. Long blocks should be better used for large data objects: full text fragments, multimedia objects, long strings, etc. Short blocks are more suitable for numerical values, strings of moderate length, date and time. It is important to consider block size of operation system, it should be multiple to the block size of data warehouse. Small block size is more suitable for systems of online transaction processing, because if server blocks data at the level of blocks, then more users can operate without interrupting each other. In the decision systems, for which more important is not general capacity (number of transactions per unit time) but average response time, larger block is more preferable.

Indices. In data warehouses the most common are indices of two types: cluster and noncluster. A feature of cluster index is that one index element corresponds to data page. A page is a physical block for data storage. When server determines a page, it determines the required record in the page. Cluster index executes the mechanism of index consecutive access, and the index is made in the form of binary tree. Since cluster index allows to reach the first record of result and to arrange subsequent search for other records without return to index level, cluster indices are suitable for data interval sampling. This category includes external keys, lists of names and any other sets of data with consecutive keys. Due to the structure of cluster indices, the search in them is faster than upon sampling of random records for respective noncluster indices. Though, this is conventional rule depending on the number of pointers required for indices of each type and on their width. Sometimes cluster index operates significantly faster and sometimes does not provide gain in the speed, though for most requests the cluster indices operate faster than the noncluster indices. Noncluster index executes mechanism of index random access. Its important feature is that one index element is accounted for one record. Identifier is assigned to each record in table.

Since in noncluster index each pointer corresponds to one record, as well as since their pointers have larger size than that of cluster indices because they contain record identifier and not just page identifier, noncluster indices occupy significantly more space. In order to increase response rate, cluster index should be created prior to noncluster ones. Creation of cluster indices requires for physical sorting of records, and noncluster indices contain pointers to pages and records, which are modified upon transfer of records. During creation of cluster index, free space in data warehouse is required in the amount of sorted copy with index, or about 120-150% of table size.

Stored procedures and functions. A stored procedure is an object of data warehouse, a set of compiled commands: server-based subprogram. Since a stored procedure is preliminary compiled and optimized, it is more efficient than similar client-based set of commands. The stored procedure is called by name as a software module, it can receive parameters and return results. The code of stored procedure can include not only operations of data extraction and modification but also branching logics, variables and other language instructions, which make the stored procedures a powerful tool of implementation of processing logics. There are four types of stored procedures:

- system,
- local,
- time,
- remote.

This classification is based on visibility. System procedures are stored in database and begin with specialized prefix. In order to modify the system stored procedure, it is required to make its copy, to save original procedure under another name. Local stored procedures are created in user databases. If a procedure is created with # or ## name, then a temporary stored procedure is created in separate database. The procedures starting from ## are global, they are visible for any communication session with this server.

The procedures with one # are local and visible only from communication where they were created. When the communication, where a procedure is created, is closed, the procedure is automatically deleted. Remote procedures can be called not only from current server communicated by client application. Herewith, the current server is an intermediate unit. There is one more type of stored procedures: expanded stored procedures.

They are cardinally different regarding the mentioned ones. These are files of DLL, generally written in high level programming language, and are referred to as stored only because the name is stored in DBMS. The DLLs are stored as files in respective list of operation system.

Triggers. Trigger is an object of data warehouse, which is a special type of stored procedure which, for instance, is executed by SQL Server during operations in this table. Triggers are executed after application of rules, default values, they provide restriction of integrity at the level of strings, reference integrity in data warehouse. Execution of any modification command results in trigger activation.

Group of requests. Requests to data warehouse are formed using SQL and inform DBMS about necessity to perform certain action. For instance, SQL requests allow as follows:

- to create table;
- to receive final data;
- to add data;
- to remove data;
- to update (modify) data;
- to protect data.

A group of requests to data warehouse contains data about activated attributes of tables for each request for information sampling, as well as frequency of each request for selected time period.

The procedure is based on statistical analysis of group of requests used by application software upon operation with relational database. Lifecycle of any stable information system, which stores data in relational data warehouse, at certain stage allows to form final set of typical requests for data reading.

Formalization of optimization of data model in the form of set-theoretical model

Construction of suboptimal data model of information system includes suboptimal distribution of tables of relational data warehouse by blocks on hard disk. The main optimization criterion of data model of information system using data warehouse is the minimum size of table string of relational data warehouse, allowing to store more data in one block and, as a consequence, to minimize the number of reading operations of data blocks from hard disk upon requests to data warehouse.

This is achieved by decrease in memory size indirectly participating in the request. Conventional flowchart of data block reading to execute request for data reading is illustrated in Fig. 1[3].

[3] S. I. Rodzin, Teoriya prinyatiya reshenii: lektsii i praktikum: Guidebook (Taganrog: TTI YuFU, 2010).

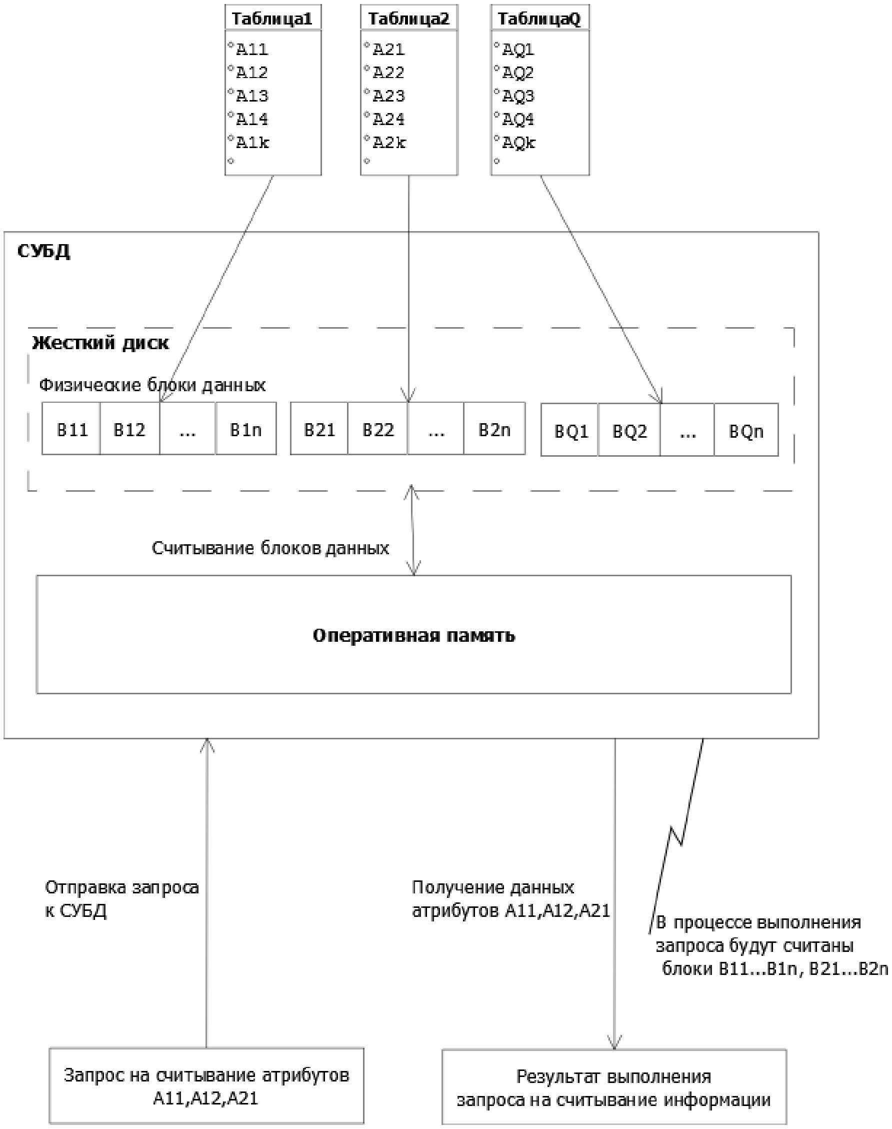


Таблица1	Table1
Таблица2	Table2
ТаблицаQ	TableQ
СУБД	DBMS
Жесткий диск	Hard disk
Физические блоки данных	Physical data blocks
Считывание блоков данных	Reading data blocks
Оперативная память	RAM
Отправка запроса к СУБД	Request to DBMS
Получение данных атрибутов A11,A12,A21	Getting data attributes A11,A12,A21
В процессе выполнения запроса будут считаны блоки B11...B1n, B21...B2n	During the request the blocks B11...B1n, B21...B2n will be read
Запрос на считывание атрибутов A11,A12,A21	Request for reading attributes A11,A12,A21
Результат выполнения запроса на считывание информации	Result of request for data reading

Figure 1
Conventional flowchart of data block reading to execute request for data reading

In the frames of the procedure it is proposed to subdivide the tables of data warehouse into several entities interrelated by one to one ratio. According to the principles of block storage of data, each table will be stored in a separate block. Upon request for information sampling, data warehouse reads data block from hard disk to RAM for each table, the attributes of which participate in the request. The flowchart of data block reading to execute request for data reading using vertical subdivision of tables is illustrated in Fig. 2

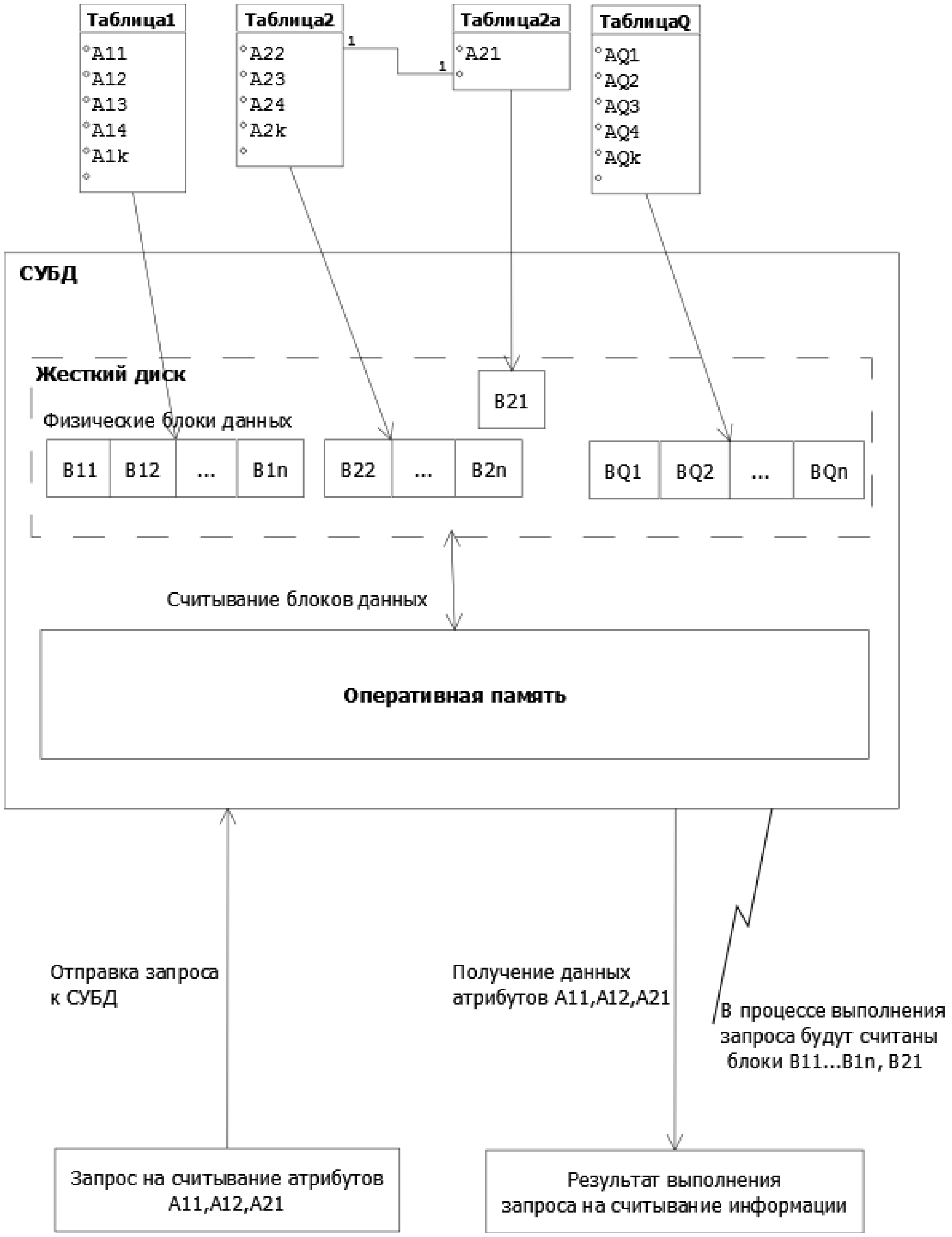


Таблица1	Table1
Таблица2	Table2
Таблица2а	Table2a
ТаблицаQ	TableQ
СУБД	DBMS
Жесткий диск	Hard disk
Физические блоки данных	Physical data blocks
Считывание блоков данных	Reading data blocks
Оперативная память	RAM
Отправка запроса к СУБД	Request to DBMS
Получение данных атрибутов A11,A12,A21	Getting data attributes A11,A12,A21
В процессе выполнения запроса будут считаны блоки B11...В1П, В21	During the request the blocks B11...B1n, B21...B2n will be read
Запрос на считывание атрибутов A11,A12,A21	Request for reading attributes A11,A12,A21
Результат выполнения запроса на считывание информации	Result of request for data reading

Figure 2
Flowchart of data block reading to execute request for data reading using vertical subdivision of tables

The performance improvement of information system is reduced to optimization of one of key subsystems: data warehouse.

In order to formalize the problem, let us consider sets and parameters affecting the processing rate of requests to the considered table of database.

1. Integer parameter TS equaling to the number of columns in the table.

2. Vector of data types $DBT = \{dbt_{idbt} \middle| idbt = \overline{1, ndbt}\}$, which are supported by selected data warehouse. Vector element: data size in bytes occupied by vector of the type.

3. Set of attributes (table columns) TA preset by binary matrix, the element of which $ta_{ita,jta}$ equals to one if the type of table column ita is jta, $ita = 1, \dots, TS, jta = 1, \dots, ndbt$.

4. For formalization of the problem let us consider the set representing the group of requests $Q = \{q_{iq} | iq = \overline{1, nq}\}$ to obtain data from database table; 2-tuple $q_{iq} = \{SFQ_{iq}, QA_{iq}\}$, where SFQ_{iq} is the numerical parameter equaling to the frequency of request per selected time interval, $QA_{iq} = \{qa_{iqa} | iqa = \overline{1, TS}\}$ is the binary vector, the dimensionality of which equals to the number of table attributes TS. $qa_{iqa} = 1$ if the table attribute TA participates in the request, otherwise, 0. nq is the number of requests.

5. Set of indices characterized by the set of table fields, used for creation of the index $IN = \{in_{iin} | iin = \overline{1, n\imath n}\}$. The set element $in_{iin} = \{in_{iin,jin} | jin = \overline{1, TS}\}$ is the binary vector, the dimensionality of which equals to the number of table attributes TS, $in_{iin,jin} = 1$ if the attribute jin of table TA participates in the index in_{iin}, otherwise, 0.

6. Stored procedures and functions $PF = \{pf_{ipf} \big| ipf = \overline{1, npf}\}$ characterized by the set of fields, used in the body of stored procedure or function. The set element $pf_{ipf} = \{pf_{ipf,jpf} | jpf = \overline{1, TS}\}$ is the binary vector, the dimensionality of which equals to the number of table attributes TS, $pf_{ipf,jpf} = 1$ if the attribute jpf of table TA participates in the body of stored procedure or function pf_{ipf}, otherwise, 0.

7. Set of triggers of database $TG = \{tg_{itg} | itg = \overline{1, ntg}\}$ characterized by the set of table fields used in the trigger body. The set element $tg_{itg} = \{tg_{itg,jtg} | jtg = \overline{1, TS}\}$ is the binary vector, the dimensionality of which equals to the number of table attributes TS, $tg_{itg,jtg} = 1$ if the attribute jtg of table TA participates in the trigger body tg_{itg}, otherwise, 0[4]

The influence of number of physical data blocks used by table of data warehouse on total number of executions of group of requests

A set of requests Q to the considered table is processed by DBMS in the time $T(Q, TA, DBT)$. The time consumption $T(Q, TA, DBT)$ can be presented in the form of sum of time consumptions for reading data blocks of tables $T_h(Q, TA, DBT)$, participating in requests Q, and other time consumptions $T_o(Q, TA, DBT)$, including time consumptions for execution of plan of request processing, for data transfer, etc.

$$T(Q, TA, DBT) = T_h(Q, TA, DBT) + T_o(Q, TA, DBT)$$

In the frames of the procedure it is proposed to decrease the term influencing total execution time of request $T_h(Q, TA, DBT)$. The time consumptions $T_h(Q, TA, DBT)$ in general form depend on the number of reading operations of data blocks in tables from hard disk. Let the time delay related with reading of one data block be T_b, then:

$$T_h(Q) = \left(\sum_{iq}^{nq} B\big(q_{iq}, TA, DBT\big) \right) * T_b,$$

where $B\big(q_{iq}, TA, DBT\big)$ is the number of data blocks to be read from hard disk to cash of data warehouse for further execution of request q_{iq} to the table preset by binary matrix TA. The cash of data warehouse is in RAM of computing device; T_b is the time delay related with reading of one data block.

The function $B\big(q_{iq}, TA, DBT\big)$ is calculated as follows:

$$B\big(q_{iq}, TA, DBT\big) = \frac{RC * RS(TA, DBT) * q_{iq}(SFQ)}{DB},$$

where RC is the number of strings in the considered table.

$RS(TA, DBT) = RSS(TA, DBT) + RST(TA, DBT),$

[4] I. V. Belchenko y R. A. Dyachenko, “Metodika povysheniya proizvoditel'nosti...

where $RS(TA,DBT)$ is the value characterizing disk space occupied by one table string in bytes; $RSS(TA,DBT)$ is the memory size occupied by service marks of DBMS for string. In particular case, it equals to the size of primary key identifier. $RST(TA,DBT)$ is the memory size occupied by table attributes in string:

$$RST(TA,DBT) = \sum\nolimits_j^{TS}\left[\left(\sum\nolimits_{idbt}^{ndbt} DBT_{idbt} * TA_{j,idbt}\right)\right],$$

where DB is the fixed size of fata block of selected data warehouse. In most data warehouses it equals to 8 Kb.

The parameters RC and DB are unchanged.

Since the time delay T_b, related with reading of one data block, can be assumed constant, the sum of time delays for reading of data blocks of table TA, $T_h(Q,TA,DBT)$, depends on the number of blocks required for reading, it is calculated as the function $F(Q,TA,DBT)$:

$$F(Q,TA,DBT) = \sum_{iq}^{nq} B\left(q_{iq},TA,DBT\right)$$

Let us substitute into $F(Q,TA,DBT)$ the equation $B\left(q_{iq},TA,DBT\right)$. The function determining the number of blocks required for reading from hard disk in RAM upon execution of requests Q to the considered table TA is as follows:

$$F(Q,TA,DBT) = \sum_{iq}^{nq}\left(\frac{RC * RS(TA,DBT) * q_{iq}(SFQ)}{DB}\right)$$

Procedure of reduction of number of physical data blocks used by data warehouse for execution of requests Q to table TA by means of its subdivision into daughter tables

In the frames of the procedure it is proposed to subdivide the considered table into $NB \in [1;TS]$ daughter tables interrelated with the parent table by 1:1 ratio.

Let us introduce the following variable:

$$x_{ij} = \begin{cases} 1, if\ the\ attribute\ j\ should\ be\ allocated\ into\ i - th\ table, \\ \qquad\qquad\qquad\qquad otherwise, 0 \end{cases}$$

The variable is the binary $TS \times TS$ matrix for table of relational database, where TS is the number of table attributes. The matrix strings correspond to the tables, into which the parent table is subdivided, and the columns correspond to their attributes. The variable corresponding to table subdivision into three daughter tables is graphically exemplified in Fig. 3.

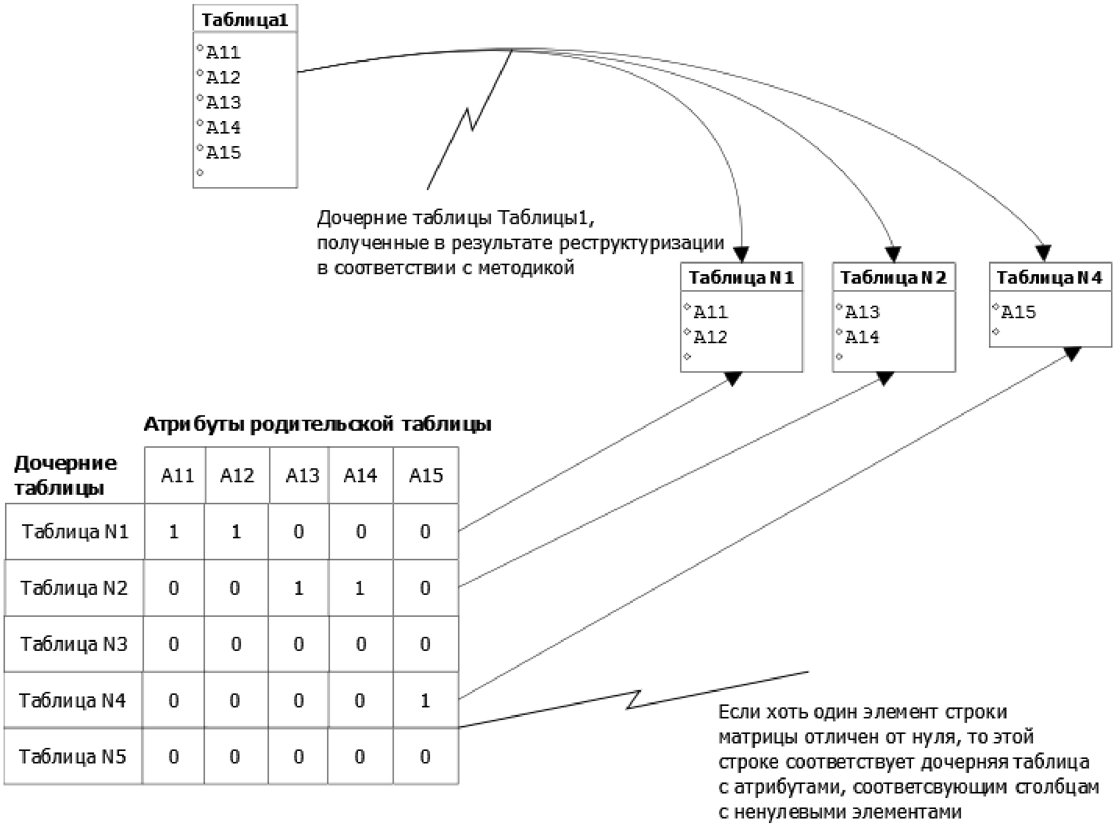


Таблица1
A11
A12
A13
A14
A15
Дочерние таблицы Таблицы1, полученные в результате реструктуризации в соответствии с методикой
Таблица N 1
A11
A12
Таблица N 2
A13
A14
Таблица N 4
A15
Атрибуты родительской таблицы
Если хоть один элемент строки матрицы отличен от нуля, то этой строке соответствует дочерняя таблица с атрибутами, соответсвующим столбцам с ненулевыми элементами

Дочерние таблицы	A11	A12	A13	A14	A15
Таблица N1	1	1	0	0	0
Таблица N2	0	0	1	1	0
Таблица N3	0	0	0	0	0
Таблица N4	0	0	0	0	1
Таблица N5	0	0	0	0	0

Таблица1	Table1
Дочерние таблицы Таблицы1, полученные в результате реструктуризации в соответствии с методикой	Daughter tables of Table1, obtained by restructuring according to the procedure
Таблица N1	Table N1
Таблица N2	Table N2
Таблица N4	Table N4
Атрибуты родительской таблицы	Attributes of parent table
Дочерние таблицы	Daughter tables
Таблица N1	Table N1
Таблица N2	Table N2
Таблица N3	Table N3
Таблица N4	Table N4
Таблица N5	Table N5
Если хоть один элемент строки матрицы отличен от нуля, то этой строке соответствует дочерняя таблица с атрибутами, соответствующим столбцам с ненулевыми элементами	If at least one element of matrix string is nonzero, then this string corresponds to daughter table with attributes corresponding to columns with nonzero elements

Figure 3
Graphic examples of variable corresponding to table subdivision into three daughter tables

Then the number of blocks read from data warehouse $BM = f(Q, RC, DB, DBT, TA, X)$, which should be read from hard disk to execute requests Q to table TA, is calculated as the function equaling to sum of blocks, which should be read from hard disk to execute requests Q to each daughter table. Maximum amount of daughter tables equals to the number of attributes NB of parent table TA.

$$\begin{aligned} BM = f(Q,RC,DB,DBT,TA,X) = \\ \sum_{iq=1}^{nq} \left\lceil \frac{RC * RSM(DBT,TA,X_1) * FQ(q_{iq},X_1)}{DB} \right\rceil + \cdots \\ + \sum_{iq=1}^{nq} \left\lceil \frac{RC * RSM(DBT,TA,X_{irs}) * FQ(q_{iq},X_{irs})}{DB} \right\rceil + \cdots \\ + \sum_{iq=1}^{nq} \left\lceil \frac{RC * RSM(DBT,TA,X_{nb}) * FQ(q_{iq},X_{nb})}{DB} \right\rceil, \end{aligned}$$

where:

$$RSM(DBT,TA,X_{irs}) = \left(\sum_{j}^{TS} \left[x_{irs,j} * \left(\sum_{idbt}^{ndbt} DBT_{idbt} * TA_{j,idbt} \right) \right] + RDS(DBT,TA,X_{irs}) \right),$$

$$FQ(q_{iq},X_{irs}) = \begin{cases} q_{iq}(SFQ), if \ \sum_{u}^{TS} (X_{irs})_u * \ (q_{iq}(QA))_u, > 0 \\ \qquad otherwise, 0 \end{cases}$$
$$irs = 1, \ldots, nb, j = 1, \ldots, TS, idbt = 1, \ldots, ndbt.$$

The parameters RC and DB are constant, RSM is the function characterizing disk space occupied by one string of daughter table irs in bytes, $RDS(DBT,TA,X_{irs})$ is the function characterizing disk space occupied by service marks of data warehouse in string of daughter table irs in bytes[5].

Therefore, the performance improvement of information system using data warehouse is reduced to search for such subdivision of table into daughter tables, at which the sum blocks to be read into cash of data warehouse for execution of requests Q is minimum.

The objective function is as follows:

$$\sum_{iq,irs} \left\lceil \frac{\left(\left(\sum_j^{TS} \left[x_{irs,j} * \left(\sum_{idbt}^{ndbt} DBT_{idbt} * TA_{j,idbt} \right) \right] + RDS(DBT,TA,X_{irs})) \right) \right) * RC * F\left(q_{iq},X_{irs}\right)}{DB} \right\rceil$$

where:

$$FQ(q_{iq},X_{irs}) = \begin{cases} q_{iq}(SFQ), if \ \sum_{u}^{TS} (X_{irs})_u * \ (q_{iq}(QA))_u, > 0 \\ \qquad otherwise, 0 \end{cases}$$
$$irs = 1, \ldots, nb, j = 1, \ldots, TS, idbt = 1, \ldots, ndbt.$$

Under structural restrictions:

[5] I. V. Belchenko y R. A. Dyachenko, “Metodika povysheniya proizvoditel'nosti…

1. Each attribute of parent table can be present only in one daughter table

$$\sum_{i_1} x_{m_1,i_1} = 1,\ m_1 = 1, \dots, TS, i_1 = 1, \dots, TS$$

2. The table attributes used upon construction of indices should belong to at least one daughter table.

$$\forall ix, ix = 1, \dots, |IN|: \prod_{m_2}^{TS} \left[\sum_{i_2}^{TS} \left(x_{m_2,i_2} * IN_{ix,i_2} - IN_{ix,i_2} \right) \right] = 0,$$
$$m_2 = 1, \dots, TS, i_2 = 1, \dots, TS.$$

3. The table attributes used in body of stored procedures or functions should belong to at least one daughter table.

$$\forall px, px = 1, \dots, |PF|: \prod_{m_3}^{TS} \left[\sum_{i_3}^{TS} \left(x_{m_3,i_3} * PF_{px,i_3} - PF_{px,i_3} \right) \right] = 0,$$
$$m_3 = 1, \dots, TS, i_3 = 1, \dots, TS.$$

4. The table attributes used in operation of triggers of the considered table should belong to at least one daughter table.

$$\forall tx, tx = 1, \dots, |TG|: \prod_{m_4}^{TS} \left[\sum_{i_4}^{TS} \left(x_{m_4,i_4} * TG_{tx,i_4} - TG_{tx,i_4} \right) \right] = 0,$$
$$m_4 = 1, \dots, TS, i_4 = 1, \dots, TS.$$

5. The ratio of the number of physical data blocks required for data storage of the considered table before the use to the number of blocks required for data storage in the daughter tables after the use of the procedure should not exceed preset parameter $TSIZE, TSIZE \in (0; 1]$.

$$TSIZE = \left({RC * RS(TA, DBT)} \Big/ {\textstyle\sum_{irs}^{nb} RC * RSM(DBT, TA, X_{irs}),} \right) / DB$$
$$irs = 1, \dots, nb$$

Search for optimum subdivision of table TA into daughter tables for execution of requests Q by exhaustive search in solution space

The objective function is nonlinear, the constraints are also nonlinear. The variable X is the $TS * TS$ binary matrix. Let us present the variable in the form of $TS * TS$ binary vector. Hence, the number of possible combinations of the variable is defined as 2^{TS*TS}. On this basis, the problem is characterized by exponential complexity and is NP difficult. A set of N variables, each of them can acquire K possible states, can have K^N possible states. Analysis of such system requires for processing of at least K^N bytes of data. A problem becomes transcomputational if $K^N > 10^{93}$.

This problem becomes transcomputational at the number of attributes $TS = 17$, then, the exhaustive search of solutions in acceptable time is unavailable[6].

The work analyzes the reduction of possible combinations of variable by constraining maximum possible number of daughter tables used for the subdivision. With this aim, it is required to reduce the number of strings in binary matrix X. Such approach would allow to determine optimum subdivision of the parent table TA into daughter tables for execution of requests Q if the number of attributes in the parent table is small. This is suitable for reference tables described by small number of attributes.

Mathematical formulation of cluster analysis based on alpha quasi-equivalence

The following definitions are given in[7]:

Definition 1. Common (nonfuzzy) n-ary ratio R between the sets $X_1, X_2, \dots, X_n$ is the subset of Cartesian product $X_1 \times X_2 \times \dots \times X_n$:

$$R \subseteq X_1 \times X_2 \times \dots \times X_n.$$

Definition 2. Fuzzy n-ary ratio R between subsets $X_1, X_2, \dots, X_n$ is the fuzzy set R, where $\forall(x_1, x_2, \dots, x_n) \in X_1 \times X_2 \times \dots \times X_n, \mu_R(x_1, x_2, \dots, x_n) \in [0,1]$;

$$X_1 = \{x_1\}, X_2 = \{x_2\}, \dots, X_n = \{x_n\}\ are \text{ regular sets.}$$

Definition 3. Fuzzy binary ratio R between the sets X, Y is the fuzzy set R, where

$\forall(x, y) \in X \times Y$,

$\mu_R(x, y) \in [0,1], X = \{x\}, Y = \{y\}$ are the regular sets. If the sets are finite, $X = \{x_1, x_2, \dots, x_n\}, Y = \{y_1, y_2, \dots, y_m\}$, then the fuzzy binary ratio R can be defined by the matrix of ratio R, the strings and columns of which are associated with the set elements X, Y, and the cross section of the $i-th$ string and the $j-th$ column is the element $\mu_R(x_i, y_j)$. Therefore,

$$R = \begin{pmatrix} \mu_R(x_1, y_1). & \cdots & \mu_R(x_1, y_m). \\ \vdots & \ddots & \vdots \\ \mu_R(x_n, y_1). & \cdots & \mu_R(x_n, y_m). \end{pmatrix}$$

Definition 4. Fuzzy binary ratio *R* on the set X is the fuzzy set R, where $\forall(x, y) \in X \times X, \mu_R(x, y) \in [0,1]$.

Definition 5. Fuzzy binary ratio R_1 belongs to fuzzy binary ratio R_2 if for $\forall x \in X, \forall y \in Y$

$$\mu_{R_1}(x, y) \leq \mu_{R_2}(x, y)$$

[6] I. V. Belchenko y R. A. Dyachenko, "Metodika povysheniya proizvoditel'nosti…

[7] D. A. Vyatchenin, Nechetkie metody avtomaticheskoi klassifikatsii: Monograph (Minsk: Tekhnoprint, 2004).

Definition 6. Fuzzy binary ratio R is referred to as reflective if $\mu_R(x,x) = 1$, $\forall x \in X$

Definition 7. Fuzzy binary ratio R is referred to as symmetrical if $\mu_R(x,y) = \mu_R(y,x)$, $\forall x, y \in X$

Fuzzy binary ratios can be presented as combination of regular binary ratios arranged by inclusion and presenting hierarchical combination of ratios. Expansion of fuzzy binary ratios is based on the notion of α level of fuzzy binary ratio.

Definition 8. α level of fuzzy binary ratio R is the regular ratio R_α defined for each α as follows[8]:

$$R_\alpha = \{(x,y) \in X \times Y | \mu_R(x,y) \geq \alpha\}$$

According to the theorem of decomposition, fuzzy binary ratio R can be presented as follows:

$$R = \bigcup_{\alpha \in [0,1]} \alpha R_\alpha$$

or

$$\mu_R(x) = \bigcup_{\alpha \in [0,1]} \left(\alpha R_\alpha(x)\right),$$

where:

$$\mu_R(x) = \begin{cases} 1, \mu_R(x) \geq \alpha \\ 0, \mu_R(x) < \alpha \end{cases}$$

Heuristic algorithm of search for suboptimal subdivision of table based on cluster analysis of requests for data warehouses of large information systems

In order to solve the problem, the procedure was developed based on cluster analysis of requests to database.

Heuristic analysis is comprised of the following stages[9]:

1) Let us consider binary fuzzy ratio R on the set $ATR = \{atr_1, atr_2, \ldots, atr_{TS}\}$ defined in the form of membership function $R = \left\{\mu_R\left(atr_{a_1}, atr_{a_2}\right) \middle| atr_{a_1}, atr_{a_2} \in ATR\right\}$, where:

$$\forall \mu_R\left(atr_{a_1}, atr_{a_2}\right) = \sum_{iq}^{nq} ((q_{iq}(QA))_{a_1} * \left(q_{iq}(QA)\right)_{a_2} * q_{iq}(SFQ)),$$

[8] V. A. Atroshchenko; V. Y. Belchenko; I.V. Belchenko y R.A. Dyachenko, “Development and research of statistical methods and optimization algorithms of search for solutions in intelligence automated systems”, International Journal of Pharmacy and Technology Vol: 8 num 2 (2016): 14137-14149.

[9] I. V. Belchenko, “Metodika povysheniya proizvoditel'nosti krupnykh informatsionnykh sistem za schet restrukturizatsii dannykh na osnove klasternogo analiza statistiki zaprosov”, CloudofScience Vol: 5 num 2 (2018): 352-366.

$$iq = 1, \ldots, nq, a_1, a_2 \in [1, TS]$$

The membership function defines in which requests the attributes atr_{a_1}, atr_{a_2} are met together. The ratio R is characterized by the following properties:

1. It is reflexive:

$$\mu_R\left(atr_{a_1}, atr_{a_1}\right) = 1, \forall\, atr_{a_1} \in\ ATR,$$

2. It is symmetric:

$$\mu_R\left(atr_{a_1}, atr_{a_2}\right) = \mu_R\left(atr_{a_2}, atr_{a_1}\right), \forall\, atr_{a_1}, atr_{a_2} \in\ ATR.$$

The algorithm of automatic clustering requires that the ratio R is characterized by transitivity $(max - min)$:

$$\mu_R\left(atr_{a_1}, atr_{a_2}\right) \geq \bigvee_{atr_k \in ATR} \left(\mu_R\left(atr_{a_1}, atr_k\right) \wedge \mu_R\left(atr_k, atr_{a_2}\right)\right),$$

$$\forall\, atr_{a_1}, atr_{a_2}, atr_k \in\ ATR.$$

$(max - min)$ is transitive closure of binary fuzzy ratio R on the set ATR, where $card(ATR) = TS$ is the binary fuzzy ratio $\breve{R}$ on the set ATR determined as follows[10,11]:

$$\breve{R} = R^1 \cup R^2 \cup \ldots \cup R^{TS},$$

where the ratios R^{TS} are determined recursively:

$$R^1 = R, R^{TS} = R^{TS-1} \circ R, TS = 2{,}3, \ldots.$$

2) After obtaining of binary fuzzy transitive ratio $\breve{R}$ the α levels are determined, $\alpha \in [0{,}1]$, the clusters $A^l, l \in [1, TS]$ are determined in accordance with the rule: if $\mu_{\breve{R}}\left(atr_{a_1}, atr_{a_2}\right) \geq \alpha$ for some $atr_{a_1}, atr_{a_2} \in ATR$, then the objects $atr_{a_1}, atr_{a_2} \in ATR$ belong to the cluster A^l;

3) From the obtained hierarchy of subdivisions, the subdivision is selected, the efficiency of which is estimated by the objective function obtained in this work. The attributes belonging to cluster are separated to single daughter table[12].

Calculation by the procedure for a table of 5 attributes. A set of requests is presented in the form of Table 1.

[10] I. V. Belchenko, “Metodika povysheniya proizvoditel'nosti...

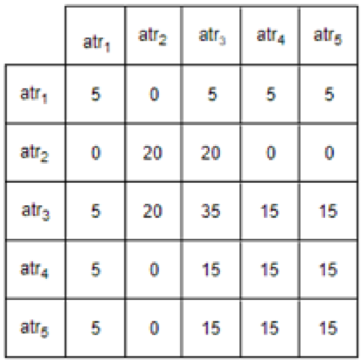
[11] A. A. Barsegyan; M. S. Kupriyanov; I. I. Kholod; M. D. Tess y S. I. Elizarov, Analiz dannykh i protsessov: Guidebook (St Petersburg: BKhV-Peterburg, 2009)

[12] I. V. Belchenko, “Metodika povysheniya proizvoditel'nosti...

No.	Number of requests to server per selected time period	Binary vector of attributes participating in request
1.	20	<0,0,1,1,0,0>
2.	5	<1,0,1,1,1>
3.	10	<0,0,1,1,1>

Table 1
Characteristics of requests

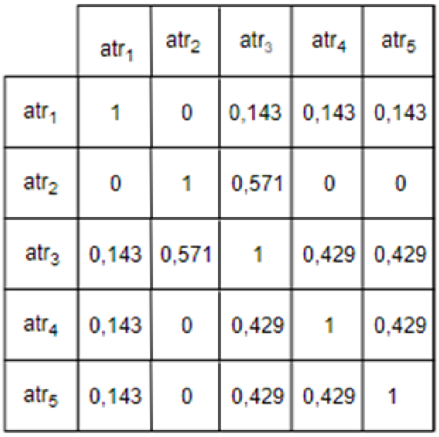
Binary ratio is constructed on the basis of a set of requests. The element of ratio equals to the frequency of occurrence of element pair in group of requests, summarized in Fig. 4.



	atr$_1$	atr$_2$	atr$_3$	atr$_4$	atr$_5$
atr$_1$	5	0	5	5	5
atr$_2$	0	20	20	0	0
atr$_3$	5	20	35	15	15
atr$_4$	5	0	15	15	15
atr$_5$	5	0	15	15	15

Figure 4
Binary ratio on the basis of request set is arranged

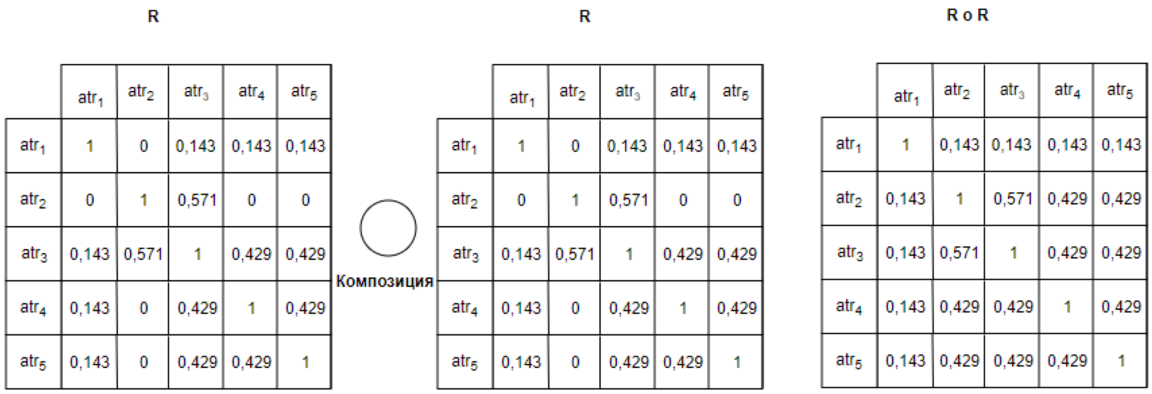
After normalization by maximum element of ratio the binary fuzzy ratio was obtained, illustrated in Fig. 5.



	atr$_1$	atr$_2$	atr$_3$	atr$_4$	atr$_5$
atr$_1$	1	0	0,143	0,143	0,143
atr$_2$	0	1	0,571	0	0
atr$_3$	0,143	0,571	1	0,429	0,429
atr$_4$	0,143	0	0,429	1	0,429
atr$_5$	0,143	0	0,429	0,429	1

Figure 5
Binary fuzzy ratio

According to the procedure, the transitive closure of fuzzy binary ratio is obtained. The result of composition of two fuzzy ratios is illustrated in Fig. 6.



R

	atr_1	atr_2	atr_3	atr_4	atr_5
atr_1	1	0	0,143	0,143	0,143
atr_2	0	1	0,571	0	0
atr_3	0,143	0,571	1	0,429	0,429
atr_4	0,143	0	0,429	1	0,429
atr_5	0,143	0	0,429	0,429	1

Композиция

R

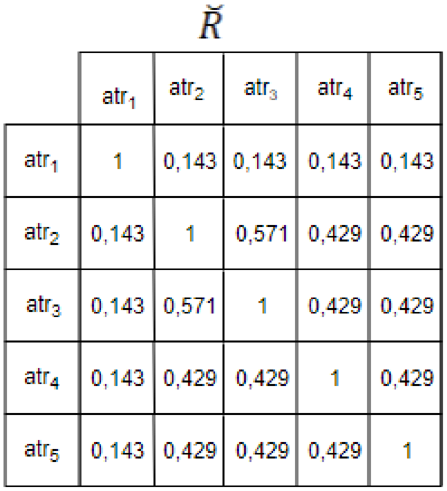
	atr_1	atr_2	atr_3	atr_4	atr_5
atr_1	1	0	0,143	0,143	0,143
atr_2	0	1	0,571	0	0
atr_3	0,143	0,571	1	0,429	0,429
atr_4	0,143	0	0,429	1	0,429
atr_5	0,143	0	0,429	0,429	1

R o R

	atr_1	atr_2	atr_3	atr_4	atr_5
atr_1	1	0,143	0,143	0,143	0,143
atr_2	0,143	1	0,571	0,429	0,429
atr_3	0,143	0,571	1	0,429	0,429
atr_4	0,143	0,429	0,429	1	0,429
atr_5	0,143	0,429	0,429	0,429	1

Figure 6
Composition of fuzzy binary ratios

Transitive closure of fuzzy binary ratio is illustrated in Fig. 7.



$\breve{R}$

	atr_1	atr_2	atr_3	atr_4	atr_5
atr_1	1	0,143	0,143	0,143	0,143
atr_2	0,143	1	0,571	0,429	0,429
atr_3	0,143	0,571	1	0,429	0,429
atr_4	0,143	0,429	0,429	1	0,429
atr_5	0,143	0,429	0,429	0,429	1

Figure 7
Transitive closure of fuzzy binary ratio

The alpha levels and clustering are summarized in Table 2.

Elements of alpha level	Value of alpha level	Value of objective function (Number of blocks read from data warehouse)
$\{A_1, A_2, A_3, A_4, A_5\}$	1	252,000
$\{A_2, A_3\}, \{A_1, A_4, A_5\}$	0.571	195,000
$\{A_1\}, \{A_2, A_3, A_4, A_5\}$	0.429	237,000
$\{A_1, A_2, A_3, A_4, A_5\}$	0.143	252,000

Table 2
Calculation of alpha levels and clustering

With the constraint, if $\mu_{\breve{R}}\left(atr_{a_1}, atr_{a_2}\right) = 1$ for certain $atr_{a_1}, atr_{a_2} \in ATR$, then the objects $atr_{a_1}, atr_{a_2} \in ATR$ belong to one cluster.

The procedure algorithm is illustrated in Fig. 8.

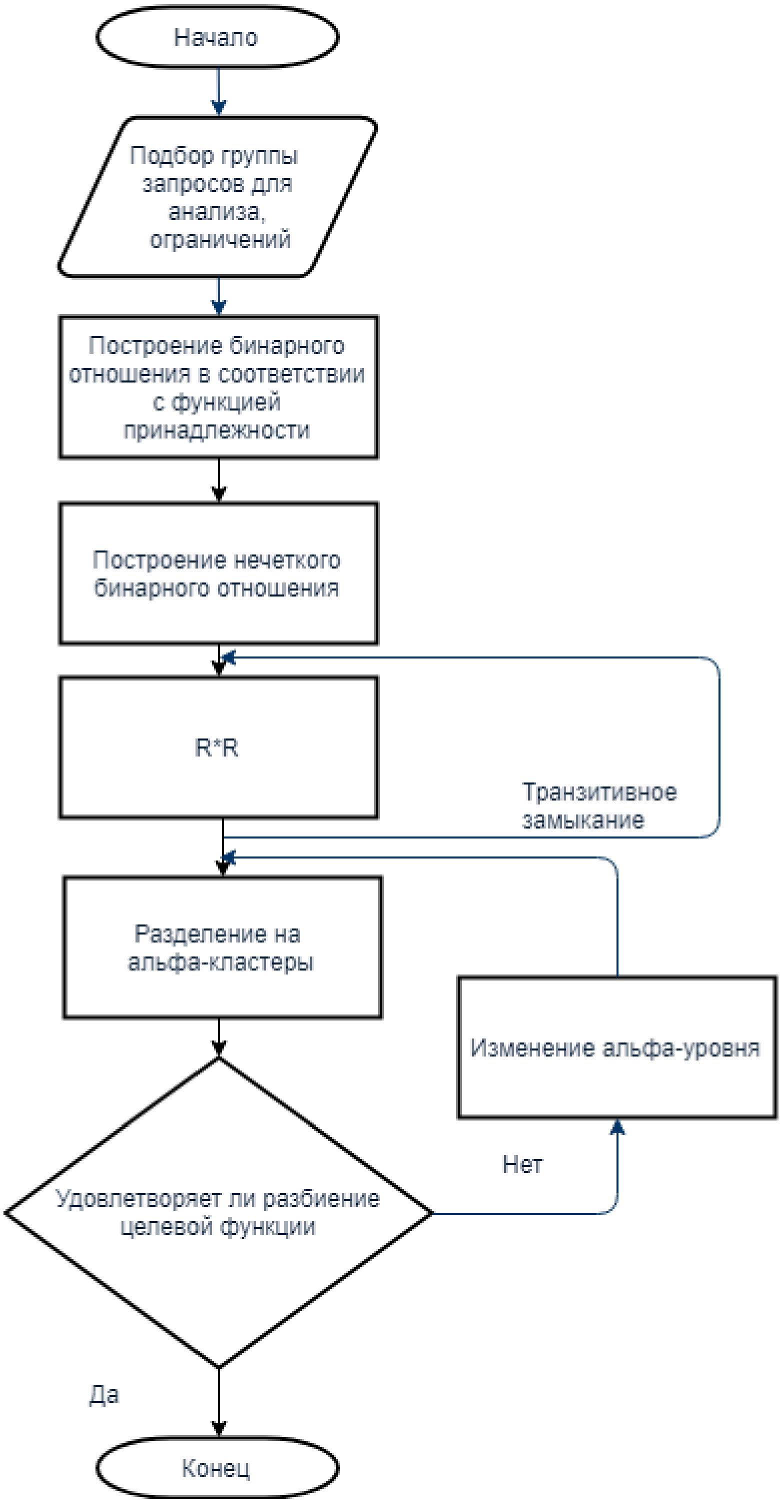

Начало
Подбор группы запросов для анализа, ограничений
Построение бинарного отношения в соответствии с функцией принадлежности
Построение нечеткого бинарного отношения
R*R
Транзитивное замыкание
Разделение на альфа-кластеры
Изменение альфа-уровня
Нет
Удовлетворяет ли разбиение целевой функции
Да
Конец

Figure 8
Algorithm of procedure

Начало	Start
Подбор группы запросов для анализа, ограничений	Selection of group of requests for analysis, constraints
Построение бинарного отношения в соответствии с функцией принадлежности	Construction of binary ratio in accordance with the membership function
Построение нечеткого бинарного отношения	Construction of fuzzy binary ratio
Транзитивное замыкание	Transitive closure
Разделение на альфа-кластеры	Subdivision into alpha clusters
Изменение альфа-уровня	Modification of alpha level
Удовлетворяет ли разбиение целевой функции	Partition satisfies objective function
Нет	No
Да	Yes
Конец	End

Development of procedure of suboptimal restructuring of table structures of data warehouses of moderate information systems on the basis of multimodal distribution of attributes

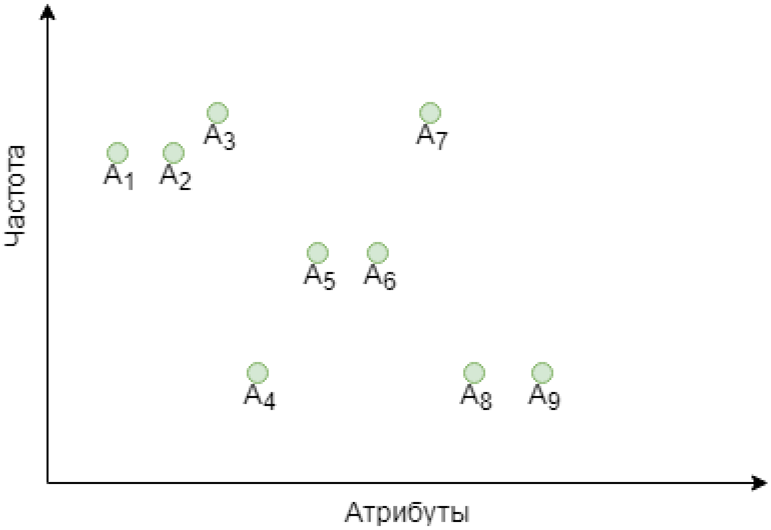
In order to solve the problem, the procedure was developed based on multimodal distribution of attributes of the considered table by their occurrence in requests for data reading. In order to obtain optimum subdivision of the considered table with the number of attributes equaling to TS, it is required[13]:

1. To obtain for each attribute the frequency of occurrence for each attribute in requests to data warehouse. The vector of occurrence frequency of attributes of the considered table in requests Q, where $\mathrm{FTA} = \{\mathrm{fta}_{\mathrm{ifta}} | \mathrm{ifta} = \overline{1,\mathrm{TS}}\}$, is

$$\mathrm{fta}_{\mathrm{ifta}} = \sum_{\mathrm{iq}=1}^{\mathrm{nq}} \mathrm{q}_{\mathrm{iq}}(\mathrm{SFQ}) * \left(\mathrm{q}_{\mathrm{iq}}(\mathrm{QA})\right)_{\mathrm{ifta}},$$
$$\mathrm{iq} = 1, \ldots, \mathrm{nq}.$$

The distribution of attributes by frequency of their occurrence in request group is exemplified in Fig. 9.

[13] I. V. Belchenko y R. A. Dyachenko, "Metodika povysheniya proizvoditel'nosti nebol'shikh informatsionnykh sistem za schet optimal'noi restrukturizatsii dannykh na osnove mnogomodal'nogo raspredeleniya atributov", Vestnik Novosibirskogo gosudarstvennogo universiteta. Seriya: Informatsionnye tekhnologii Vol: 16 num 2 (2018): 19-30.



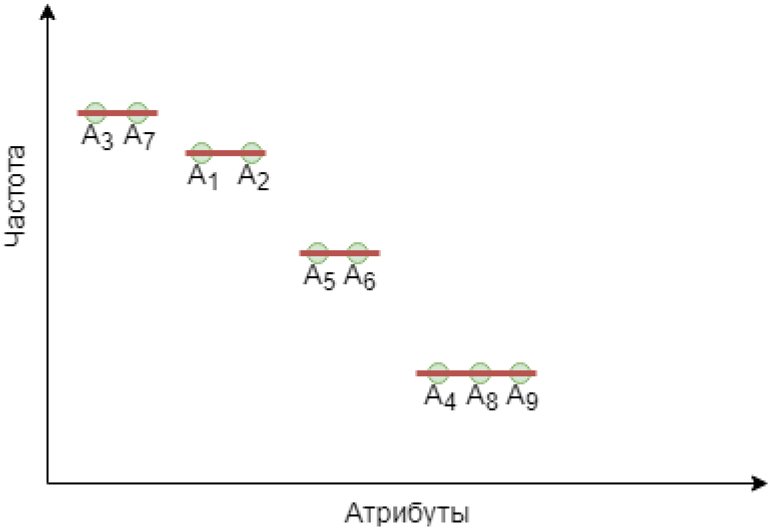
Частота	Frequency
Атрибуты	Attributes

Figure 9
Graphical example of distribution of attributes by frequency of their occurrence in request group

1. To sort the attributes by the frequency of their occurrence in requests. $\mathrm{FTA}' = \{\mathrm{fta}'_{\mathrm{ifta}} | \mathrm{ifta} = \overline{1,\mathrm{TS}}, \mathrm{fta}' \in \mathrm{FTA}, \mathrm{fta}'_{\mathrm{ifta}} \le \mathrm{fta}'_{\mathrm{ifta}+1}\}$,

2. To form groups of attributes with the same frequency. We obtain the partition of final set FTA'. $\mathrm{GF} = \{\mathrm{gf}_1, \dots, \mathrm{gf}_{\mathrm{bn}}\}$, where $\mathrm{gf}_1 \cup \dots \cup \mathrm{gf}_{\mathrm{bn}} = \mathrm{FTA}', \mathrm{gf}_{\mathrm{bi}} \neq \emptyset, 1 \le \mathrm{bi} \le \mathrm{bn}, \text{where } \forall\ \mathrm{ifta}, \mathrm{fta}'_{\mathrm{ifta}} \in \mathrm{gf}_{\mathrm{bi}}, \text{if } \forall\ (\mathrm{gf}_{\mathrm{bi}})_{\mathrm{bj}} = \mathrm{fta}'_{\mathrm{ifta}}, \mathrm{ifta} = \overline{1,\mathrm{TS}}, \mathrm{bj} = \overline{1, |\mathrm{gf}_{\mathrm{b\imath}}|}$.

Attribute grouping by the frequency of their occurrence in requests is exemplified in Fig. 10.

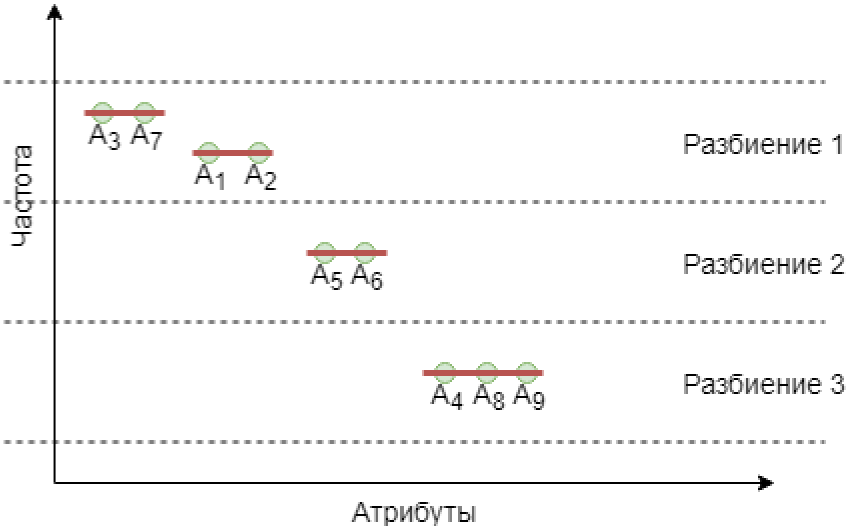


Частота	Frequency
Атрибуты	Attributes

Figure 10
Graphic example of attribute grouping by frequency of their occurrence in request group

3. To obtain set partition of attribute group. We obtain partition of final set GF. $\mathrm{GF}' = \{\mathrm{gf}'_1, \dots, \mathrm{gf}'_{\mathrm{gn}}\}$, where $\mathrm{gf}'_1 \cup \dots \cup \mathrm{gf}'_{\mathrm{gn}} = \mathrm{GF}, \mathrm{gf}'_{\mathrm{gi}} \neq \emptyset, 1 \leq \mathrm{gi} \leq \mathrm{gn}, \text{where} \, \forall \; \mathrm{gf}_{\mathrm{bi}}, \mathrm{gf}_{\mathrm{bi}} \in \mathrm{gf}'_{\mathrm{gi}}$,

$\frac{(\mathrm{gi}-1)*\mathrm{TS}}{\mathrm{K}} \leq \ \mathrm{bi} < \frac{\mathrm{gi}*\mathrm{TS}}{\mathrm{K}}, \ 1 \leq \mathrm{bi} \leq \mathrm{bn}.\mathrm{K}$ is the coefficient characterizing the number of partitions of attribute groups, $\mathrm{K} \in [1, \mathrm{gn}]$. Subdivision of the considered table into daughter tables is exemplified in Fig. 11.



Частота	Frequency
Атрибуты	Attributes
Разбиение 1	Partition 1
Разбиение 2	Partition 2
Разбиение 3	Partition 3

Figure 11
Graphic example of considered table subdivision into daughter tables.

By varying the parameter $\mathrm{K} \in [1, \mathrm{gn}]$ characterizing the number of partitions of a set of attribute groups, the procedure allows to obtain solutions, the efficiency of which is estimated by the objective function developed in the studies[14].

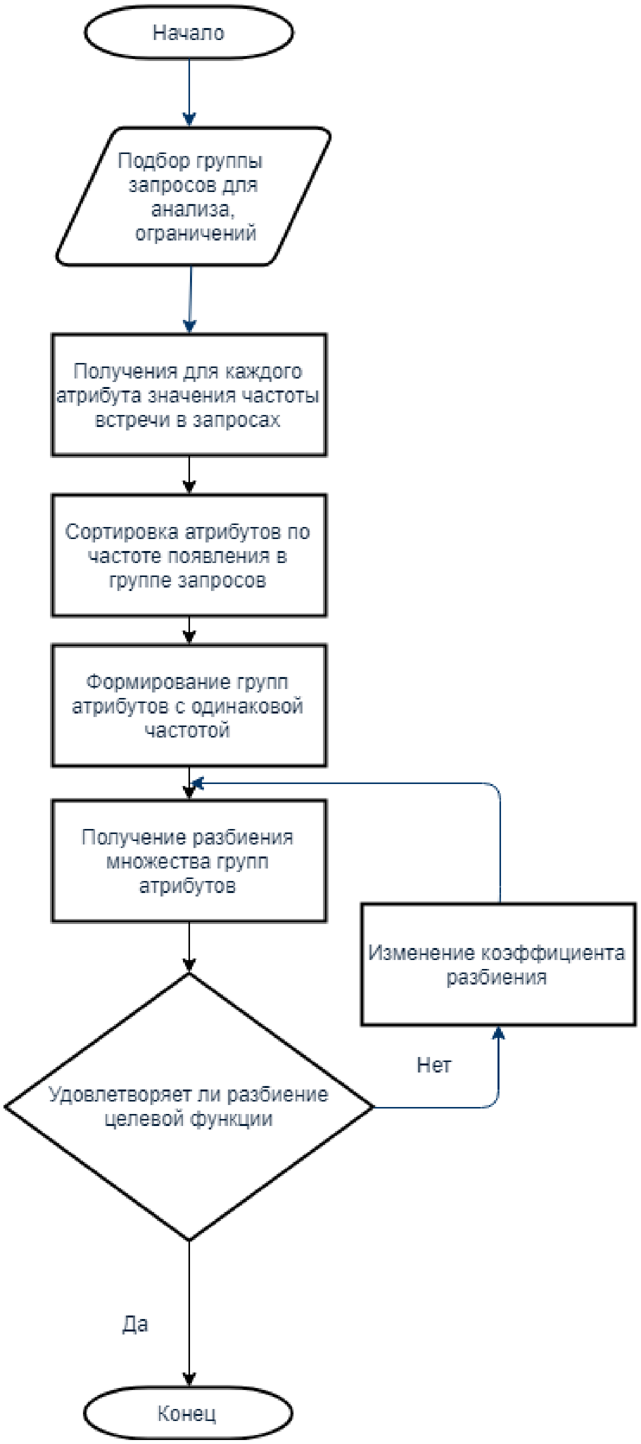
Calculation by the procedure for a table of 5 attributes. Prediction of the objective function and consideration for attributes of the table are omitted and described in Section 4. A set of requests is presented in the form of Table 3.1. The obtained frequency of occurrence of each attribute in the group of requests to data warehouse is summarized in Table 3.

[14] I. V. Belchenko y R. A. Dyachenko, “Metodika povysheniya proizvoditel'nosti...

fta_1	fta_2	fta_3	fta_4	fta_5
5	20	35	15	15

Table 3
Frequency of attribute occurrence in group of requests

The algorithm of the procedure is illustrated in Fig. 12.


Начало
Подбор группы запросов для анализа, ограничений
Получения для каждого атрибута значения частоты встречи в запросах
Сортировка атрибутов по частоте появления в группе запросов
Формирование групп атрибутов с одинаковой частотой
Получение разбиения множества групп атрибутов
Изменение коэффициента разбиения
Нет
Удовлетворяет ли разбиение целевой функции
Да
Конец

Начало	Start
Подбор группы запросов для анализа, ограничений	Selection of group of requests for analysis, constraints
Получения для каждого атрибута значения частоты встречи в запросах	Getting occurrence frequency in requests for each attribute
Сортировка атрибутов по частоте появления в группе запросов	Sorting of attributes by occurrence frequency in requests
Формирование групп атрибутов с одинаковой частотой	Formation of attribute groups with the same frequency
Получение разбиения множества групп атрибутов	Obtaining partition of a set of attribute groups
Изменение коэффициента разбиения	Modification of partition coefficient
Удовлетворяет ли разбиение целевой функции	Partition satisfies objective function
Нет	No
Да	Yes
Конец	End

Figure 12
Algorithm of the procedure application

Conclusion

1. Domain area of data warehouses has been described. The main definitions of objects of data warehouses, their structures, interrelations, influence on performance have been presented.

2. The influence of data block size of physical model of data warehouse on its performance has been determined.

3. Optimization of data warehouse has been formalized, including description of decision space, variables, main parameters, objective function, restrictions.

4. Nonlinear type of objective function has been determined, impossibility to solve the problem by exhaustive search methods in acceptable time has been determined due to its dimension. A method to decrease the problem dimension has been proposed.

5. Heuristic algorithm has been proposed to search for suboptimal table subdivision, based on cluster analysis of request group for data warehouses of large information systems.

6. The procedure of suboptimal restructuring of table structures for data warehouses of moderate information systems has been developed on the basis of multimodal attribute distribution.

References

Atroshchenko, V. A.; Belchenko, V. Y.; Belchenko, I. V. y Dyachenko, R. A. Development and research of statistical methods and optimization algorithms of search for solutions in intelligence automated systems”. International Journal of Pharmacy and Technology Vol: 8 num 2 (2016): 14137-14149.

Barsegyan, A. A.; Kupriyanov, M. S.; Kholod, I. I.; Tess, M. D. y Elizarov, S. I. Analiz dannykh i protsessov: Guidebook. St Petersburg: BKhV-Peterburg. 2009.

Belchenko, I. V. “Metodika povysheniya proizvoditel'nosti krupnykh informatsionnykh sistem za schet restrukturizatsii dannykh na osnove klasternogo analiza statistiki zaprosov”. CloudofScience Vol: 5 num 2 (2018): 352-366.

Belchenko, I. V. y Dyachenko, R. A. “Metodika povysheniya proizvoditel'nosti informatsionnoi sistemy za schet optimal'noi restrukturizatsii dannykh. Izvestiya vysshikh uchebnykh zavedenii. Povolzhskii region”. Tekhnicheskie nauki num 1 (2018): 26-38.

Belchenko, I. V. y Dyachenko, R. A. “Metodika povysheniya proizvoditel'nosti nebol'shikh informatsionnykh sistem za schet optimal'noi restrukturizatsii dannykh na osnove mnogomodal'nogo raspredeleniya atributov”. Vestnik Novosibirskogo gosudarstvennogo universiteta. Seriya: Informatsionnye tekhnologii Vol: 16 num 2 (2018): 19-30.

Chigarina, E. I. Databases: guidebook. Samara: SGAU. 2015.

Rodzin, S. I. Teoriya prinyatiya reshenii: lektsii i praktikum: Guidebook. Taganrog: TTI YuFU. 2010.

Vyatchenin, D. A. Nechetkie metody avtomaticheskoi klassifikatsii: Monograph. Minsk: Tekhnoprint. 2004.

REVISTA INCLUSIONES M.R.
REVISTA DE HUMANIDADES
Y CIENCIAS SOCIALES

CUADERNOS DE SOFÍA
EDITORIAL

Las opiniones, análisis y conclusiones del autor son de su responsabilidad
y no necesariamente reflejan el pensamiento de **Revista Inclusiones**.

La reproducción parcial y/o total de este artículo
debe hacerse con permiso de **Revista Inclusiones.**