УДК 658.5:629.78
Doi: 10.31772/2712-8970-2021-22-4-589-599

**Для цитирования:** Картамышев А. С., Черныш Б. А., Мурыгин А. В. Формирование многомерных данных в информационной финансово-экономической системе на предприятии госкорпорации «Роскосмос» // Сибирский аэрокосмический журнал. 2021. Т. 22, № 4. С. 589–599. Doi: 10.31772/2712-8970-2021-22-4-589-599.

**For citation:** Kartamyshev A. S., Chernysh B. A., Murygin A. V. Method for forming multi-dimensional data in the information financial and economic system at the enterprise of state space corporation "Roscosmos". *Siberian Aerospace Journal.* 2021, Vol. 22, No. 4, P. 589–599. Doi: 10.31772/2712-8970-2021-22-4-589-599.

# Формирование многомерных данных в информационной финансово-экономической системе на предприятии госкорпорации «Роскосмос»

А. С. Картамышев[1*], Б. А. Черныш[2], А. В. Мурыгин[2]

[1]АО «Информационные спутниковые системы» имени академика М. Ф. Решетнева»
Российская Федерация, 662972, г. Железногорск Красноярского края, ул. Ленина, 52
[2]Сибирский государственный университет науки и технологий имени академика М. Ф. Решетнева
Российская Федерация, 660037, г. Красноярск, просп. им. газ. «Красноярский рабочий», 31
[*]E-mail: kartam@iss-reshetnev.ru

*Построение детализированного учета, позволяющего генерировать сложную, аналитическую отчетность, является непреложным требованием современной финансовой системы. Для предприятий госкорпорации «Роскосмос» с позаказным и мелкосерийным производством, работающих в условиях 275 ФЗ, отличающихся большим количеством НИОКР и высокой степенью неопределенности в процессе создания продукции, эта задача является системообразующей. Отчетность должна содержать согласованные непротиворечивые данные на любом участке управленческого и бухгалтерского учета на любой момент времени. Наряду с этим, система должна обеспечивать гибкость, надежность и быстродействие, присущие транзакционным базам данных. Для построения информационной поддержки, удовлетворяющей указанным условиям, требуется либо разделять OLTP и OLAP схемы данных, либо применять специализированные решения, основанные на использовании структур и методик, оптимизированных для выполнения OLAP операций в традиционных РСУБД. В данной статье рассматривается подход к формированию многомерных данных в автоматизированной системе управления финансово-экономическими задачами как эффективная альтернатива сложным и дорогостоящим BI-решениям. В отличие от многих коммерческих решений, описываемая система не хранит избыточные данные (например, регистры оперативного учета в платформе «1С: Предприятие»), необходимые для построения аналитического учета. Лежащие в ее основе структуры данных и методы их обработки позволяют осуществлять все виды учета и иметь мощные инструменты построения аналитической отчетности. В статье предлагаются алгоритмы работы системы на примере построения простых OLAP кубов, применяемых в реальных задачах автоматизации финансово-экономической деятельности в АО «ИСС» для одной из подсистем «Покупки». Проведена формализация этих задач, рассмотрен математический аппарат построения многомерных моделей данных на основе информации из фиксированного набора нормализованных таблиц реляционной БД. Представлены примеры SQL запросов и выходных данных. Обобщены преимущества применения системы в оперативном, управленческом и бухгалтерском учете на предприятии, повышающие ее эксплуатационную эффективность.*

*Ключевые слова: база данных, OLTP, OLAP, аналитическая обработка, многомерное представление, реляционная схема, нормализация, нормальная форма, система автоматизации.*

# Method of forming of multidimensional data in information financial and economic system at the State Corporation "Roscosmos" enterprise

A. S. Kartamyshev[1*], B. A. Chernysh[2], A. V. Murygin[2]

[1]JSC Academician M. F. Reshetnev Information Satellite Systems
52, Lenin St., Zheleznogorsk, Krasnoyarsk region, 662972, Russian Federation
[2]Reshetnev Siberian State University of Science and Technology
31, Krasnoyarskii rabochii prospekt, Krasnoyarsk, 660037, Russian Federation
[*]E-mail: kartam@iss-reshetnev.ru

*The construction of detailed accounting that allows to generate complex analytical reporting is an indispensable requirement of a modern financial system. For enterprises of the State Space Corporation "Roskosmos" with custom-made and small-scale production, operating in the conditions of Federal Law No. 275, characterized by a large amount of R&D and a high degree of uncertainty in the process of creating products, this task is a system-forming one. The reporting should contain consistent data in any area of management and accounting at any given time. Along with this, system must provide the flexibility, reliability and performance inherent in transactional databases. To build information support that satisfies the specified conditions, it is required either to separate OLTP and OLAP data schemes, or to apply specialized solutions based on the use of structures and techniques optimized for performing OLAP operations in traditional RDBMSs. This article considers the approach to form the multidimensional data in an automated management system for economic tasks, as an effective alternative to complex and expensive BI solutions. Unlike many commercial systems, the ASU FEZ does not store redundant data (for example, operational accounting registers in the 1C: Enterprise platform) required to build analytical accounting. The underlying data structures and methods of their processing allow for all types of accounting and have powerful tools for constructing analytical reporting. The article proposes algorithms for the operation of the system using the example of building simple OLAP cubes used in real tasks of automating financial and economic activities in ISS JSC for one of the Purchase subsystems. The formalization of these problems is carried out, the mathematical apparatus for constructing multidimensional data models based on information from a fixed set of normalized tables of a relational database is considered.The examples of SQL queries and outputs are provided. The advantages of using the system in operational management and accounting at the enterprise increasing its operational efficiency are summarized.*

*Keywords: database, OLTP, OLAP, analytical processing, multidimensional representation, relational schema, normalization, normal form, automation system.*

## Introduction

One of the main requirements for financial and economic systems is the ability to quickly obtain relevant analytical information needed to make operational management decisions and the generation of comprehensive detailed reporting. This task is especially urgent for the enterprises of the state corporation Roskosmos, which operate under the state defense order and are obliged to keep separate accounting of costs for various projects in their economic activities. Information support for accounting and management accounting is based on the rules for storing and processing data from primary source documents registered in accounting systems, their connections and the processes of forming new information accounting objects. Data in different areas of accounting is nothing more than derivatives of primary documents or a set of primary information. In the process of redistribution and consumption of data in certain areas of accounting, information links between various derivatives from one base may be lost. Information objects begin to "live their own lives" and generate new derivatives. With information gaps, the possibility of differentiating the current derivative to the primary source of data (foundation) and analyzing data related to it within the framework of technological and economic processes is lost. The mathematical solution of such problems, leading to a logically built chain of operations and the possibility of accounting automation, is the subject of research by many scientists and practitioners working on the creation of information support systems for financial and economic tasks in enterprise management.

**Research Analysis**

The leading place in solving this class of problems is occupied by OLAP technologies [1] (OLAP - online analytical processing, online analytical processing). Systems that implement OLAP must comply with the so-called FASMI principles [2]: *Fast* (fast system response), *Analysis* (full-featured analysis), *Shared* (multi-user access to data), *Multidimensional* (multidimensional conceptual representation of data), *Information* (obtaining information in the right volume where needed). As you can see, based on these principles, the requirements for a reporting system, which can really be called an OLAP system, are very strict. The reports should be built on millions of records in seconds, just as quickly disclosed by the analysts of interest to the user, filtered and regrouped [3].

The most common approach in the implementation of OLAP systems is the separation of data between two databases: transactional (OLTP - online transaction processing) and analytical (OLAP). At the same time, the OLTP database is optimized for fast data entry, and OLAP for the fast construction of complex reporting. In fact, an OLAP database is usually built according to a special architecture and contains pre-calculated aggregate data, which ensures high query execution speed. The price for this is the need to synchronize OLTP and OLAP. Since such a process, as a rule, is periodic, there is a delay between the appearance of data in the operational database and analytical data. This approach is implemented in almost all modern BI systems (business intelligence - collection and analysis of business information): Microsoft Analysis Services, Oracle OLAP, IBM Informix, Arbor Essbase, etc.

Another approach to solving problems of online analytical data processing is to embed OLAP tools directly into the OLTP storage. This is achieved both by using special data elements (for example, operational accounting registers in the 1C: Enterprise platform [4]) and optimal (from the point of view of optimizing the execution of OLAP queries) data schema structuring. This approach formed the basis of the financial and economic task management system developed at the enterprise (FES ACS) [5] as an effective alternative to complex and expensive BI systems.

**Approach description**

The structures and algorithms underlying the ACS FES allow you to effectively combine the flexibility and speed of a transactional database (DB), on the one hand, and the ability to build complex reporting in a reasonable time, on the other. Further, the article provides an overview of the technical implementation of the data warehouse using the example of the "Purchases" subsystem, which is responsible for accounting for goods, works and services that an enterprise acquires to create its own sales objects.

The core of the "Purchases" module consists of 5 tables built in accordance with the rules of normal forms [6; 7], and is shown in fig. one:

– POK_PRIH – receipt documents for inventory items, works, services;

– POK_OSN – basis documents for VAT accounting (incoming invoices);

– PO_SOGL – documents-grounds for payment (acceptance-consent of responsible employees to pay for inventory items, works, services);

– POK_OPL – payment documents;

– POK_SW – link table, contains links in the form of foreign keys to the primary keys of records from the tables listed above (ID_PR – to the POK_PRIH table, ID_OSN – to the POK_OSN table, ID_TS – to the POK_SOGL table, ID_OPL – to the POK_OPL table).

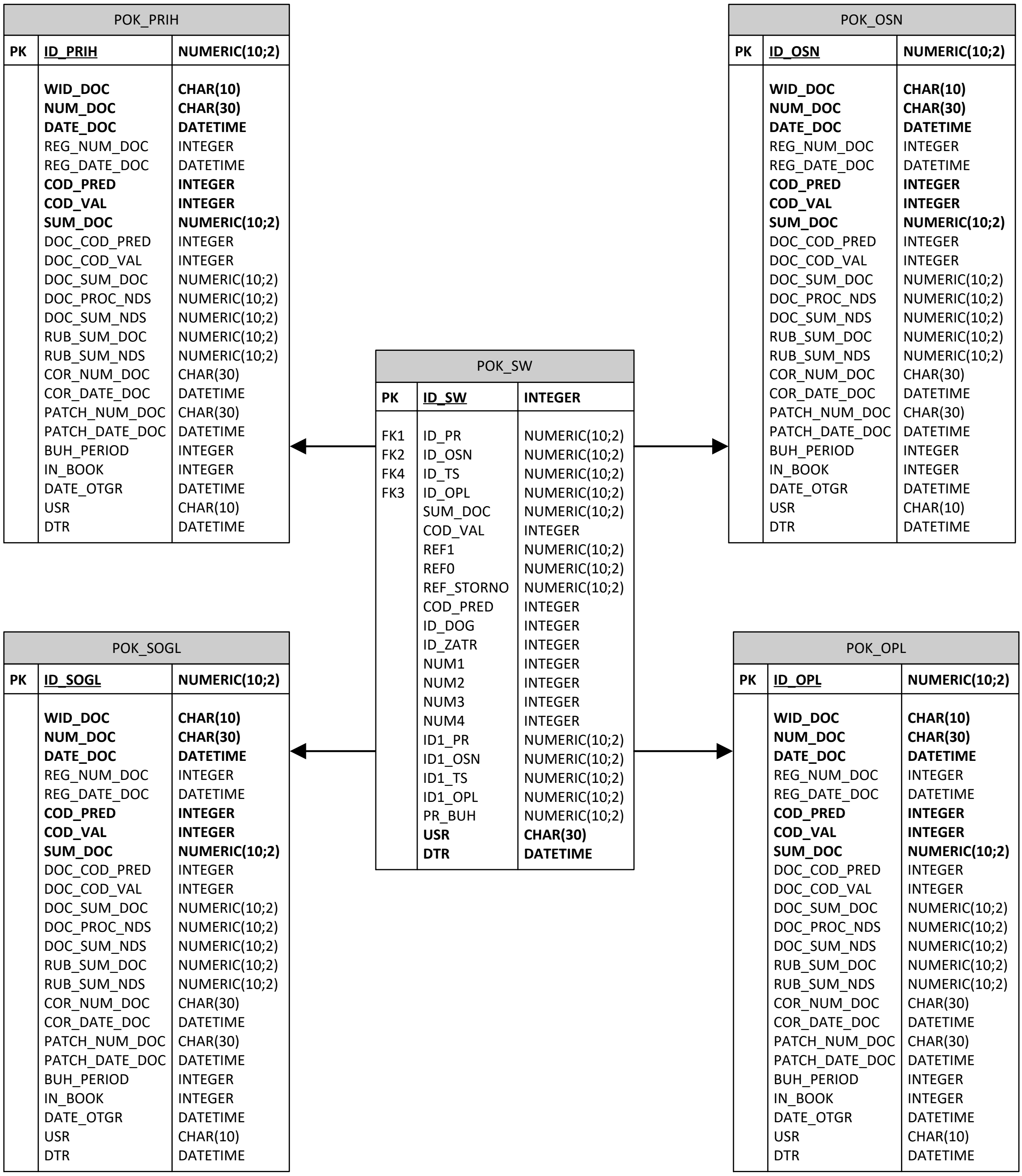


Fig. 1. Module “Purchase” data scheme

Рис. 1. Схема данных модуля «Покупки»

The proposed approach to data structuring allows you to perform a wide range of analytical processing tasks without preliminary data preparation or the use of intermediate representations. For a visual description, let us formalize them [8],then we will consider several real examples of such tasks solved within the framework of the "Purchases" subsystem. So, let the database scheme $\Re = \{R_1, R_2, ..., R_k\}$ be specified, obtained as a result of the normalization of relations [6; 7]. The relations $R_i$ are defined on the set of attributes U = {A1, A2,..., An}. Let $[R_i]$ be a relation scheme, a set of attributes on which the relation $R_i$ is defined. Suppose that the scheme $\Re$ is reduced [7], i.e. there are no two relations such that $[R_i] \subseteq [R_j]$, in $\Re$ is $i \neq j$.. Tuple $t[X]$ is a set of attribute values $A_j \in X \subseteq [R_i]$ Aj specified in tuple $t \in R_i$. The *NULL* value of attribute $t$: $t[A_j] = NULL$ is not equal to any other value, including another null value.

Многомерное представление будем задавать в виде совокупности размерностей $\{D_1, D_2, \dots, D_d\}$, где $D_l$ – множество расширенных имен атрибутов: $R_iA_j$, $A_\varphi \in [R_i]$; $M$ – множество мер, также заданных в виде расширенных имен атрибутов. Значения $D_l$ являются значениями координат гиперкуба, значения $M$ будут располагаться в рабочей области гиперкуба. Для каждой размерности задается ограничение в виде логической формулы $F_l$.

We will define a multidimensional representation as a set of dimensions $\{D_1, D_2, \dots, D_d$, where $D_l$ is the set of extended attribute names: $R_iA_j$, $A_\varphi \in [R_i]$; $M$ is a set of measures, also specified as extended attribute names. The $D_l$ values are the hypercube coordinate values, the $M$ values will be located in the hypercube workspace. For each dimension, a constraint is specified in the form of a logical formula $F_l$.

**Example 1.** It is necessary to find payment data (any transfer of funds) to an enterprise with code 12345 under project 111 and supply contracts 222, 223, where funds were paid from 01/01/2021 to 02/28/2021 according to budget planning items (BPS) 333, 334 and received goods/works/services in the period from 01/01/2021 to 01/31/2021.

In order to simplify the formulation and implementation, in this and subsequent examples, attributes not used in the selection are omitted. Thus, within the framework of the described database schema, we have the following subset of attributes: $A_1$ - connection identifier, $A_2$ - record identifier

on receipt of goods/works/services, $A_3$ – payment record identifier, $A_4$ – company code, $A_5$ – project identifier, $A_6$ – supply contract identifier, $A_7$ – SBP, $A_8$ – receipt date, $A_9$– payment date, $A_{10}$– payment amount. The following functional dependencies exist here: $DEP = \{A_1 \to A_2A_3A_4A_5A_6A_7, A_2 \to A_8, A_3 \to A_9A_{10}\}$. Based on this subset, we obtain a simplified relationship scheme: Document links = $R_1(\boldsymbol{A_1}, A_2, A_3, A_4, A_5, A_6, A_7)$, Receipts of goods/works/services= $R_2(\boldsymbol{A_2}, A_8)$ , Payment documents = $R_3(\boldsymbol{A_3}, A_9, A_{10})$, where key relationship attributes are highlighted in bold. One of the possible representations of the hypercube is given in table 1.

*Table 1*

**Payment data**

| **Enterprise code** | 12345 | | | |
|---|---|---|---|---|
| **Project** | 111 | | | |
| **Supply contract** | 222 | | 223 | |
| **BPS** | 333 | 334 | 333 | 334 |
| **Date** | *Sum* | *Sum* | *Sum* | *Sum* |
| 21.01.2021 | 75000 | 250000 | 357500 | 65500 |
| 26.01.2021 | 31000 | 310000 | 785000 | 96000 |
| 28.01.2021 | 70000 | 870080 | 6500 | 55000 |

Dimension attributes are in bold, fact attributes are in italics, and attribute values are in regular font. Diagram of the hypercube in Table. 1 can be represented as follows:

$$\{R_3.A_9\} \times \{R_1.A_4\{R_1.A_5\{R_1.A_6\{R_1.A_7(R_3.A_{10})\}\}\}\},$$

where $D_1 = \{R_3.A_9\}$ и $D_2 = \{R_1.A_4, R_1.A_5, R_1.A_6, R_1.A_7\}$ – measurements, $M = \{R_3.A_{10}\}$ – facts. Logical constraint: $F = (R_1.A_2 \neq NULL \wedge R_1.A_3 \neq NULL \wedge R_1.A_4 = 12345 \wedge R_1.A_5 = 111 \wedge (R_1.A_6 = 222 \vee R_1.A_6$

= 223) $\wedge$ ($R_1.A_7$ = 333 $\vee$ $R_1.A_7$ = 334) $\wedge$ $R_3.A_{11}$ >= 01.01.2021 $\wedge$ $R_3.A_{11}$ <= 28.02.2021 $\wedge$ $R_2.A_8$ >= 01.01.2020 $\wedge$ $R_8.A_8$ <= 31.01.2020).

The corresponding SQL query is shown in fig. 2.

```
SELECT pok.cod_pred AS pok_cod_pred, -- enterprise
  pok.id_zatr AS pok_id_zatr, -- project
  pok.id_dog AS pok_id_dog, -- contract
  pok.num4 AS pok_num4, -- budget plan item
  opl.date_doc AS opl_date_doc, -- payment document date
  opl.sum_doc AS opl_sum_doc -- payment document amount
FROM pok_sw pok
  LEFT JOIN pok_opl opl ON pok.id_opl = opl.id_opl
  LEFT JOIN pok_prih pr ON pok.id_pr = pr.id_prih
WHERE pok.id_opl IS NOT NULL -- money was paid
  AND pok.id_pr IS NOT NULL -- goods/works/services were received
  AND pok.cod_pred = 12345 -- data for the enterprise with the code
12345
  AND pok.id_zatr = 111 -- for project with code 111
  AND pok.id_dog = 222 -- under contract with code 222
  AND pok.num4 = 333 -- the payment was by SBP with code 333
  AND opl.date_doc BETWEEN TO_DATE('01/01/2021','dd.mm.yyyy') AND
TO_DATE('02/28/2021','dd.mm.yyyy') -- payment in the period
  AND pr.date_doc BETWEEN TO_DATE('01/01/2020','dd.mm.yyyy') AND
TO_DATE('01/31/2020','dd.mm.yyyy') -- the arrival was in the period
```

Fig. 2. SQL query of payments data to the enterprise with the specified analytics

Рис. 2. SQL запрос данных оплат предприятию с указанной аналитикой

Using the described database scheme DB, the request includes any other measurements and facts (data and analytics of payments, invoices, acceptances, receipts etc.) in a similar way, allowing you to track all the movement (documents, operations) on a given or other logical constrain

**Example 2.** It is necessary to find the receivables (the amount of debt) to the enterprise with code 12345 for delivered goods/works/services under projects 111 and 112.

The minimum subset of attributes is specified: $A_1$ – communication identifier, $A_2$ – record identifier for the receipt of goods/works/services, $A_3$ – payment record identifier, $A_4$ – enterprise code, $A_5$– project identifier, $A_6$ – supply contract identifier, $A_7$ – receipt document number, $A_8$ is the amount of the receipt document, $A_9$ is the reporting period. On this set of attributes, there are the following functional dependencies: $DEP = \{A_1 \rightarrow A_2A_3A_4A_5A_6, A_2 \rightarrow A_7A_8A_9\}$. We get the relationship scheme: Document links =$R_1(\boldsymbol{A_1}, A_2, A_3, A_4, A_5, A_6)$ , Receipts of goods/works/services = $R_2(\boldsymbol{A_2}, A_7, A_8, A_9)$. One of the possible representations of the hypercube is given in Table. 2.

Diagram of the hypercube in Table. 2 can be represented as follows:

$$\{R_2.A_7\} \times \{R_1.A_4\{R_1.A_5\{R_1.A_6\{R_2.A_9(R_2.A_8)\}\}\}\},$$

where $D_1 = \{R_2.A_7\}$ и $D_2 = \{R_1.A_4, R_1.A_5, R_1.A_6, R_2.A_9\}$ – measurements, $M = \{R_2.A_8\}$ – acts. Logical constraint: $F = (R_1.A_2 \neq NULL \wedge R_1.A_3 = NULL \wedge R_1.A_4 = 12345)$.

*Table 2*

**Debit debt**

| **Enterprise code** | 12345 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Project** | 111 | | | | 112 | | | |
| **Supply contract** | 222 | | 223 | | 224 | | 225 | |
| **Reporting period** | 333 | | 334 | | 333 | | 334 | |
| **№ doc.** | *I/2020* | *II/2020* | *III/2020* | *IV/2020* | *I/2020* | *II/2020* | *III/2020* | *IV/2020* |
| 11 | 75000 | 21500 | 250000 | 784200 | 357500 | 650800 | 65500 | 48000 |
| 12 | 31000 | 79200 | 310000 | 58100 | 785000 | 130000 | 96000 | 51700 |
| 13 | 70000 | 623300 | 870080 | 3700 | 6500 | 210000 | 55000 | 3580 |

An example of the corresponding SQL query is shown in fig. 3.

```
SELECT pok.cod_pred AS pok_cod_pred, -- enterprise
   pok.id_zatr AS pok_id_zatr, -- project
   pok.id_dog AS pok_id_dog, -- contract
   prih.num_doc AS prih_num_doc, -- receipt document number
   prih.sum_doc AS prih_sum_doc, -- receipt document amount
   prih.cod_val AS prih_cod_val, -- currency code of the receipt docu-
ment
   prih.buh_period AS prih_buh_period -- accounting period
FROM pok_sw pok
   LEFT JOIN pok_prih prih ON pok.id_pr = prih.id_prih
WHERE pok.id_pr IS NOT NULL -- there was a receipt of goods (the pres-
ence of a receipt document)
   AND pok.id_opl IS NULL -- there was no payment
   AND pok.cod_pred = 12345 -- data for the enterprise with the code
12345
```

Fig. 3. SQL data debit debt query on accounts for the specified company

Рис. 3. SQL запрос данных о дебиторской задолженности указанному предприятию

The result will be all lines for the enterprise under study, where primary receipt documents are registered, for all projects, supply contracts with possible payments in any currency. Here you can analyze which delivery was paid for by what, which acceptances, advances or payments were agreed, under which contracts, for whom, which invoices for tax accounting and when they should get into the books of purchases and sales, etc. An example of calculating the amount of payments to an enterprise is shown in fig. 4.

```
  SELECT SUM(DECODE(NVL(pok.id_opl, 0), 0, 0, pok.sum_doc)) AS sum_opl,
pok.cod_val
FROM pok_sw pok
WHERE pok.id_pr IS NOT NULL -- there was an arrival of goods (the presence
of a receipt order)
  AND pok.cod_pred = 12345 -- enterprise data with code 12345
GROUP BY pok.cod_val
```

Fig. 4. SQL debt amount query to the specified company

Рис. 4. SQL запрос суммы долга указанному предприятию

**Example 3.** Getting information about what goods/works/services were paid for by this document and on the basis of which agreement (acceptance) document it was done.
The minimum subset of attributes is specified: $A_1$ – connection identifier, , $A_2$ – record identifier for the receipt of goods/works/services, $A_3$ – agreement (acceptance) document record identifier, $A_4$– payment record identifier, $A_5$ – enterprise code, $A_6$ – project identifier, $A_7$ – delivery contract identifier, $A_8$– receipt document type , $A_9$ – receipt document number, $A_{10}$ – receipt amount, $A_{11}$ – acceptance number, $A_{12}$ – payment document number, $A_{13}$ – payment document date. There are the following dependencies: $DEP = \{A_1 \rightarrow A_2A_3A_4A_5A_6A_7,\ A_2 \rightarrow A_8A_9(A_{10}),\ A_3 \rightarrow A_{11},\ A_4 \rightarrow A_{12}A_{13}\}$ (three functional and one multivalued).

In this example, it is proposed to abandon the need to fulfill the functional dependence [9–13]

$$D_1D_2\ldots D_d \rightarrow M,$$

which means that any composite vector of values of dimensions $D_1D_2\ldots D_d$ corresponds to no more than one vector of values of measures M.

Rejection of this dependency will allow using meaningful (not key) attributes in dimensions and having several values (list) of an attribute in one hypercube cell $R_iA_j \in M$.. Lists of values are used in data analysis when parameter values do not need to be correlated with objects.

We get the following relationship scheme: Document links = $R_1(\boldsymbol{A_1}, A_2, A_3, A_4, A_5, A_6, A_7)$,Receipts of goods/works/services = $R_2(\boldsymbol{A_2}, A_8, A_9, A_{10})$, Payment approval documents = $R_3(\boldsymbol{A_3}, A_{11})$, Payment documents = $R_4(\boldsymbol{A_4}, A_{12}, A_{13})$.. One of the possible representations of the hypercube is given in table 3.

*Таблица 3*

**Payment document data**

| **Enterprise code** | 12345 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Project** | 111 | | | | | | | |
| **Supply contract** | 222 | | | | 223 | | | |
| **Receipt document image** | Receipt note | | Receipt order | | Certificate of completed work | | Receipt note | |
| **№ acceptance** | *Sum* | *№ Receipt document* | *Sum* | *№ Receipt document* | *Sum* | *№ Receipt document* | *Sum* | *№ Receipt document* |
| 11 | 75000 | 21 | 357500 | 24 | 250000 | 27 | 65500 | 30 |
| 12 | 31000 | 22 | 785000 | 25 | 310000 | 28 | 96000 | 31 |
| 13 | 70000 | 23 | 6500 | 26 | 870080 | 29 | 55000 | 32 |

Diagram of the hypercube in Table. 3 can be represented as follows:

$$\{R_3.A_{11}\} \times \{R_1.A_5\{R_1.A_6\{R_1.A_7\{R_2.A_8(R_2.A_9)\}(R_2.A_{10})\}\}\},$$

where $D_1 = \{R_3.A_{11}\}$ и $D_2 = \{R_1.A_5, R_1.A_6, R_1.A_7, R_2.A_8\}$ – measurements, $M = \{R_3.A_9, R_3.A_{10}\}$ – facts. Logical constrain: $F = (R_4.A_{12} = 987 \wedge R_4.A_{13} = 21.01.2021)$.

An example of the corresponding SQL query is shown in fig. 5.

The ACS module of the FEP "Sales" and other auxiliary modules are implemented in a similar way. Together, these modules form the core of the system.

The database schema was originally designed in accordance with the rules of normal forms [6; 7] and has the property of a lossless connection (LBI) in accordance with Theorem 5.8 [6]:

*Let σ be a decomposition of the relation R formed by relation schemes in third normal form, and let also X be the key of R. Then τ = σ ∪{X} is a decomposition of R such that all its constituent relation schemes are in third normal form. This decomposition preserves dependencies and preserves the lossless connection property.*

```
SELECT pok.cod_pred AS pok_cod_pred, -- enterprise
  pok.id_zatr AS pok_id_zatr, -- project
  pok.id_dog AS pok_id_dog, -- contract
  prih.wid_doc AS prih_wid_doc, -- receipt document type
  prih.num_doc AS prih_num_doc, -- receipt document number
  prih.sum_doc AS prih_num_doc, -- receipt amount
  sogl.num_doc AS osn_num_doc – acceptance number
FROM pok_sw pok
  LEFT JOIN pok_prih prih ON pok.id_pr = prih.id_prih
  LEFT JOIN pok_sogl sogl ON pok.id_osn = sogl.id_osn
  LEFT JOIN pok_opl opl ON pok.id_opl = opl.id_opl
WHERE opl.num_doc = 987 -- №  document number  from the payment regis-
tration table
  AND opl.date_doc = TO_DATE('21.01.2021','dd.mm.yyyy') – date document
from the payment registration table
```

Fig. 5. SQL query for payment data

Рис. 5. SQL запрос данных платежа

The resulting data cubes correspond to the formal rules considered in [14], in accordance with the "composite table" data model, which,in turn, is a generalization of the "semantic transformation" model [12] for the case of a list of values in one cell. This algorithm generates hierarchies in the dimensions of a hypercube using functional, multivalued dependencies of the source database and attribute hierarchies specified by the user.

The described method of storing and processing data allows you to efficiently perform both OLTP and OLAP operations without creating additional data structures and performing unnecessary transformations. This, in turn, leads to a significant reduction in overhead both in terms of the complexity of compiling queries and in terms of their execution time compared to the classic OLAP toolkit, contributes to a more optimal use of data warehouse resources, and avoids the need to store redundant data. The application of the described algorithm in OLAP systems allows to reduce the time for generating a schema of a new multidimensional data model, as well as to make the presentation of the model the most convenient for the user.

**Conclusion**

The proposed option for the formation and storage of data of primary documents on the economic activities of the enterprise provides a detailed separate accounting of financial and economic transactions of purchases and sales, and allows you to effectively solve the problems of building accounting, tax and management accounting. The use of the above schemes, the logic of storing and processing data for working with their various derivatives makes it possible to differentiate the elements of the chain of events (states) in accounting to primary sources linked in the link table in accordance with the logic of business transactions, which allows you to analyze data for any set of interest analytics. Thus, at any stage and section of accounting there are always integral information links. This provides the ability to build complex OLAP data structures in combination with the speed of a transactional DBMS. The efficiency of the FEZ ACS and its underlying algorithms is confirmed by the experience of many years of commercial operation at one of the leading enterprises of the state corporation Roscosmos JSC ISS without the involvement of external commercial solutions.

**References**

1. Vychugzhanina T. V., Dolgova I. A. [Using data warehouses and OLAP-technology in modern ERP-systems]. *Informacionnye tekhnologii v modelirovanii i upravlenii: podhody, metody, resheniya*. Penza, 2017 (In Russ.).

2. Avrunev O. E., Stasyshin V. M. *Modeli baz dannykh* [Database models]. Novosibirsk, Novosibirsk state technical university Publ., 2018 (In Russ.).

3. *Kompaniya "Abis Soft". Primenenie OLAP-tekhnologiy dlya uchetnykh sistem na platforme 1S* [Abis Soft company. Using of OLAP technologies for accounting systems on the 1C platform]. Available at: https://www.cfin.ru/itm/olap/1c.shtml (accessed: 11.02 2021).

4. Goncharova Y. A., Andreeva N. M. [Reading data from the 1C: Enterprise 8.1 periodic information register using the methods of the infobase tabular model]. *Aktual'nye problemy aviacii i kosmonavtiki*. 2012, No. 1, P. 393–394 (In Russ.).

5. Kartamyshev A. S., Vilkov Y. V. [Innovation in aerospace enterprise control. Integrated information control system of the enterprise JSC "ISS"]. *Aktual'nye voprosy proektirovaniya avtomaticheskih kosmicheskih apparatov dlya fundamental'nyh i prikladnyh nauchnyh issledovaniy*, Anapa, 2017 (In Russ.).

6. Ulman J. *Osnovy sistem baz dannykh* [Principles of Database Systems]. Moscow, Finance and statistics Publ., 1983, 334 p.

7. Maier D. The Theory of Relational Databases. Rockville, Computer Science Press, 1983.

8. Zykin S. V., Mosin S. V., Poluyanov A. N. [Technology of separate generation of multidimensional data]. *Vestnik Donskogo gosudarstvennogo tekhnicheskogo universiteta.* 2016, No. 2(85), P. 121–128 (In Russ.).

9. Lechtenborger J., Vossen G. Multidimensional normal forms for data warehouse design. *Information Systems Frontiers*. 2003, No. 28, P. 415–434.

10. Lehner W., Albrecht J., Wedekind H. Normal forms for multidimensional databases. *Proceedings of the Tenth International Conference on Scientific and Statistical Database Management*. 1998, P. 63–72.

11. Mazon J., Trujillo J., Lechtenborger J. Reconciling requirement-driven data warehouses with data sources via multidimensional normal forms. *Data & Knowledge Engineering*. 2007, No. 3 (63), P. 725–751.

12. Zykin S. V. Formation of Hypercube Representation of Relational Database. *Programming and Computer Software*. 2006, No. 6 (32), P. 348–354.

13. Zykin S. V. Dynamic contexts of relational-type database. *Informatics and Applications*. 2014, No. 1 (8), P. 77–88.

14. Redreev P. G. [Construction of Hierarchies in Multidimensional Data Models]. *Izvestiya Saratovskogo universiteta*. 2009, No. 4, P. 7–12 (In Russ.).

**Библиографические ссылки**

1. Вычугжанина Т. В., Долгова И. А. Использование хранилищ данных и OLAP-технологии в современных ERP-системах // Информационные технологии в моделировании и управлении: подходы, методы, решения : сб. тр. конф. Пенза, 2017.

2. Аврунев О. Е., Стасышин В. М. Модели баз данных. Новосибирск : Изд. НГТУ, 2018. 124 с.

3. Компания «Абис Софт». Применение OLAP-технологий для учетных систем на платформе 1С [Электронный ресурс]. URL: https://www.cfin.ru/itm/olap/1c.shtml (дата обращения: 11 02 2021).

4. Гончарова Ю. А., Андреева Н. М. Чтение данных периодического регистра сведений «1С:Предприятие 8.1» методами табличной модели информационной базы // Актуальные проблемы авиации и космонавтики. 2012. № 1. С. 393–394.

5. Картамышев А. С., Вилков Ю. В., Инновации в управлении предприятием РКО. Интегрированная информационная система управления предприятием АО «ИСС» // Актуальные вопросы проектирования автоматических космических аппаратов для фундаментальных и прикладных научных исследований. Анапа, 2017.

6. Ульман Дж. Основы систем баз данных. М. : Финансы и статистика, 1983. 334 с.

7. Maier D. The Theory of Relational Databases. Rockville: Computer Science Press, 1983.

8. Зыкин С. В., Мосин С. В., Полуянов А. Н. Технология раздельного формирования многомерных данных // Вестник Донского гос. техн. ун-та. 2016. № 2(85). С. 121–128.

9. Lechtenborger J., Vossen G. Multidimensional normal forms for data warehouse design // Information Systems Frontiers. 2003. No. 28. P. 415–434.

10. Lehner W., Albrecht J., Wedekind H. Normal forms for multidimensional databases // Proceedings of the Tenth International Conference on Scientific and Statistical Database Management. 1998. P. 63–72.

11. Mazon J., Trujillo J., Lechtenborger J. Reconciling requirement-driven data warehouses with data sources via multidimensional normal forms // Data & Knowledge Engineering, 2007. No. 3 (63). P. 725–751.

12. Zykin S.V. Formation of Hypercube Representation of Relational Database. Programming and Computer Software. 2006. No. 6 (32). P. 348–354.

13. Zykin S.V. Dynamic contexts of relational-type database. Informatics and Applications. 2014. No. 1 (8). P. 77–88.

Редреев П. Г. Построение иерархий в многомерных моделях данных // Известия Саратовского ун-та, 2009. № 4. С. 7–12.



**Картамышев Александр Сергеевич** – начальник группы автоматизации задач управления в Обществе; АО «Информационные спутниковые системы» имени академика М. Ф. Решетнева». E-mail: kartam@iss-reshetnev.ru.

**Черныш Борис Александрович** – аспирант; Сибирский государственный университет науки и технологий имени академика М. Ф. Решетнева. E-mail: borisblack@mail.ru.

**Мурыгин Александр Владимирович** – доктор технических наук, профессор, заведующий кафедрой информационно-управляющих систем; Сибирский государственный университет науки и технологий имени академика М. Ф. Решетнева. E-mail: avm514@mail.ru.

**Kartamyshev Alexandr Sergeevich** – Chief of control process automation group in the Company, JSC Academician M. F. Reshetnev "Information Satellite Systems". E-mail: kartam@iss-reshetnev.ru.

**Chernysh Boris Aleksandrovich** – postgraduate student; Reshetnev Siberian State University of Science and Technology. E-mail: borisblack@mail.ru.

**Murygin Alexandr Vladimirovich** – Dr. Sc., Professor; Reshetnev Siberian State University of Science and Technology. E-mail: avm514@mail.ru.