|  |
| --- |
| NATIONAL RESEARCH UNIVERSITY – HIGHER SCHOOL OF ECONOMICS  FACULTY oF BUSINESS INFORMATICS |
| Draft on the project |
| «Data Modeling Methods in Analytical Information Systems» |
|  |

­­­­

|  |
| --- |
|  |

Moscow, 2013

**Student:**Iserson N.M.,

group № 471

**Scientific advisor:**

Morgunov A.F.,

PhD, associate professor

**Language advisor:**

Klepko E.Y.,

senior lecturer

Content

[Abbreviations 2](#_Toc348788412)

[Introduction 3](#_Toc348788413)

[1. Problem description 4](#_Toc348788414)

[1.1 Technical problem description 4](#_Toc348788415)

[1.2 Business Case 5](#_Toc348788416)

[2. Current project solution based on SAP BPC system 8](#_Toc348788417)

[3. Proposed project solution based on SAP HANA platform 9](#_Toc348788418)

[4. Conclusion 12](#_Toc348788419)

[5. References 13](#_Toc348788420)

# Abbreviations

|  |  |
| --- | --- |
| **Обозначение** | **Описание** |
| SAP BW | SAP Business Warehouse |
| SAP BPC | SAP Business Planning and Consolidation |
| SAP HANA | SAP High-Performance Analytical Appliance |
| OLTP | Online Transaction Processing |
| OLAP | Online Analytical Processing |
| MDX | Multi-Dimensional Expressions |
| SSD | Solid-State Drive |
| ETL | Extraction, Transformation and Loading |
| EDW | Enterprise Data Warehouse |

# Introduction

To begin with, it is important to define what budget planning system is. Firstly, the budgeting system is an analytical system, or at least, a system based on the analytical platform. But secondly, it is a strategic and operational budgets data collection tool. But there are some inconsistencies that exist in the approach to modeling data schemes of large commercial budgeting systems.

It is well known that the analytical systems data schemes are based on some principles that have already become traditional. We are talking about the concept of data warehouse or OLAP which is an approach to design that focuses on fast querying of a specific type which is required in the reporting system. In OLTP systems there are other requirements: the model should work similarly for all possible types of queries, and avoid duplication of data.

Here it is necessary to remember such term as normalization. Transactional database scheme should be normalized in order to avoid anomalies like deleting, updating, reading, etc. Analytical systems schemes are denormalized or multidimensional, so it makes it possible to execute queries operatively and more specifically. Also there is both detailed and aggregated data in a multidimensional scheme, which is also known as the star-schema or cube that improves the efficiency of a query response. Usually the size of storage is much larger than the size of transactional systems.

The inconsistency of budgeting systems is that they have to combine the properties of analytical and transactional systems. Firstly, they are used to build analytical budget forms for top management. Secondly, there are transactional processing systems (update, delete), which are designed for data collection during the budget approval. Also it is necessary to monitor the amount of data, which is handled by such intricate systems, because it is often difficult to define the tactical budget and operational one as the same. Also it is important to establish the boundaries of "granularity" of data processed by the system at the design phase because the data is known to occur in the system only with launch.

# Problem description

## 1.1 Technical problem description

Here are some problems of classic budget planning system data model.

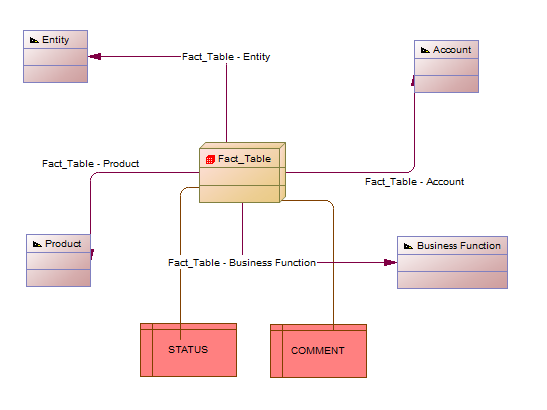
OLTP requirements:

* Data insert / update;
* Locking of data slices with statuses;
* Data and actions audit;
* Drivers of costs and exchange rates;

OLAP requirements:

* Fast analytical queries (slice and dice);
* Hierarchies maintenance and navigation;
* Business Intelligence;
* Multivariate analysis (pivoting).

According to best practices data model contains OLAP-cube or multidimensional fact table and reference tables (“dimensions”) [1]. It is also known as Star Schema Benchmark [3]:



## 1.2 Business Case

Now we will review some parts of business blueprint of budgeting system project implementation. Business case will allow us to understand the significance of problems which are described here.

**Organizational Structure**

Organizational structure of commercial bank can be represented as the hierarchy tree. Leafs are the smallest business units, where expenses have to be planned.

These business units include:

* Head Office Business Units;
* Branches in Moscow and Moscow Region;
* Head Offices of Branches (first level operational offices);
* Additional offices (second level operational offices);
* International Representative Offices;
* Affiliated companies.

**Master Data Management**

For planning purposes, following reference tables have to be implemented in budgeting system:

* Business Units;
* Accounts;
* Expenses Types;
* Exchange Rates;
* Drivers;
* Projects;
* Property.

**Document Flow and Work Statuses**

Work statuses are designed for tracking and locking data through many steps of budget approval. Business users are able to change work statuses for different perspectives: Business Units, Accounts, Time, Category, etc.

There are several work statuses:

* Data Input;
* Head Office Control;
* Financial Department Control;
* Approved;
* Rejected.

In case of rejecting records such data is fully excluded from budgeting process and all values are set to zero. Rejected records are copied to the “Rejected” category, business user who input the data gets the e-mail notification and audit data is saved in the database.

**Report “Comments”**

This report allows business users to observe all comments in the current view. Also, there are predefined filters such as “Time”, “User”, “Priority” and dimension members.

**Report “Data Audit”**

This report shows business users data changes and many attributes (time, cause, input interface). Report rows contain full information about the record lifecycle. However, in business reports only new value is shown.

**Report “Work Statuses”**

This report allows business users to define tracking parameters including date, time and dimension members, such as accounts, business units and categories. The report shows current work status of record and current step in budget approval.

In conclusion, budgeting system requirements prove importance of questions which are declared in the technical problem description. On the one hand, data model have to be optimized for manual input, tracking work statuses and comments. Usually the basis for it is a highly-normalized table scheme. On the other hand, analytical tasks require multidimensional model to perform fast slice and dice queries. In the following chapters current project solution as well as new innovative database appliance will be reviewed. According to the business case, the latter one will be our subject of interest here.

# Current project solution based on SAP BPC system

Firstly, data model of budgeting system is based on SAP BW platform. SAP BW is the SAP solution for enterprise data warehousing. It allows architects to use the following advantages during the data modeling [2]:

1. OLAP, MDX Engine – multidimensional, slice and dice analysis
2. ETL processes – transformation data flows, temporary data storage for source data (Persistent Staging Area), delta loading or Changed-Data Capture.
3. Opportunities for data modeling and performance management with analytical cubes (partitioning, compression, indexes, aggregates)

However, star-schema is not fully optimized for insert, update and delete operations and, coupled with high data granularity, it leads to slow performance [4]. The cause of this effect is frequent joining between multidimensional fact table and dimensions, work statuses and comments. Fact table is in “many-to-one” relationship with such tables, because for every record all versions of comments and work statuses have to be stored. It is the requirement for permanent data auditing.

# Proposed project solution based on SAP HANA platform

At the moment, database technologies are at the turning point. Classic approaches and abstractions are revised such as physical data storage, hardware organization and microprocessor’s architecture. The widespread relational database theory is not cure-all solution. Thus, it is necessary to use all opportunities for architecture optimization to deal with current model restrictions.

The last innovations for data storage and access in analytical systems are:

1. GPU-driven databases [5];

High parallelization of queries due to large amount of simple microprocessors, shared memory and multithreading. This technology is implemented in the Empulse ParStream DBMS.

1. In-Memory Databases;

Fast data retrieval, caching and parallelization, in-memory data storage (Random Access Memory). This technology is implemented in the SAP HANA [6], QlikView In-Memory.

1. Column-oriented databases [13];

Fast read and group operations, parallel aggregation, compression (because data is stored not row-by-row, but by columns). This technology is implemented in the SAP HANA, MongoDB, Apache Cassandra [7].

1. Special parallelization algorithms and frameworks [9]

* Map-Reduce (examples – Sybase IQ, Apache Hadoop, Google Tenzing [8])
* Parallel Aggregation, Hash tables [10], [11] (example – SAP HANA)

1. Distributed or eventually-consistent systems [12];

They provide high availability due to duplicated data storage. Any data copy is available for read operations, while insert and update operations could be performed consequentially during the certain time period. This technology is implemented in the Apache Cassandra, Amazon Dynamo.

In this chapter the demo project solution for budgeting system based on SAP HANA platform will be described. This product combines such innovative technologies as column-oriented storage, in-memory data storage and high query parallelization.

SAP HANA is an «appliance», i.e. the combination of hardware and software components. It comes pre-installed on a specific appliance hardware system delivered in conjunction with certified partners (IBM, Dell, Cisco, Fujitsu).

Thus, the SAP HANA has the following advantages:

1. In-Memory Storage
2. Column-oriented Storage
3. Non-materialized aggregates
4. Parallel Aggregation, Map-Reduce Frameworks

Our research is aimed at building innovative data model combining advantages of row and column oriented storages:

1. Data input will be performed using highly-normalized tables with row-based storage
2. Analytical queries will use column-oriented table

The latter one will be the implementation of Google Big Table approach [14], or, in other words, fact-table without dimensions. It is possible because a column in the column-oriented table can be added without affecting current data. Furthermore, joining and indexing are not necessary anymore which leads to the better query performance results.

Data model scheme is shown in the picture:

e

The comparison of query performance acceleration is the main issue of this work. The same amount of data will be processed by comparable hardware machines.

The expected result is the strong performance acceleration in SAP HANA. It leads to an opportunity to perform real-time analytical queries. Besides, the system will be available even in case of large amount of insert and update operations during budget approval process. Different model configurations, for instance, partitioning, indexes and metadata will allow us to make multivariate analysis of SAP HANA advantages.

# Conclusion

In the framework of budgeting data model problems, a model that meets the requirements of both transactional input and online analytical processing was built.

The expected result is a real-time query performance as well as preconditions for distributed and parallelized processing of data input.

Despite the fact that the model is tied to the details of the physical realization and requires the implementation of the latest technologies at a low level, it should not be considered as a drawback. The database technologies are now experiencing a major transformation that, together with increasing amounts of data, will affect the physical architecture of the database in the nearest future. That is why there are certain bases, which at a certain stage of development should be revised. What is more, the concept of distributed computing studied in this research enables to achieve scale-out performance, and data model established meets all the requirements of the input, modification and deletion.

# References

[1] Kimball, Ralph. «The data warehouse ETL toolkit: practical techniques for extracting, cleaning, conforming, and delivering data».

[2] Amol Palekar, Bharat Patel, and Shiralkar Shreekant. «SAP NetWeaver BW 7.3 — Practical Guide (2nd Edition)».

[3] Patrick O'Neil, Elizabeth (Betty) O'Neil, Xuedong Chen and Steve Revilak. «The Star Schema Benchmark and Augmented Fact Table Indexing, Presentation at TPCTC (Transaction Processing Performance Council Technical Conference), Lyon, France, 2009».

[4] Peter Jones. «Implementing SAP Business Planning and Consolidation».

[5] Peter Bakkum, Srimat Chakradhar. «Efﬁcient Data Management for GPU Databases».

[6] Franz Färber, Sang Kyun Cha, Jürgen Primsch, Christof Bornhövd, Stefan Sigg, Wolfgang Lehner. «SAP HANA Database - Data Management for Modern Business Applications».

[7] Venkata Ponnam, Indiana University. «A survey on Cassandra».

[8] Biswapesh Chattopadhyay, Liang Lin, Weiran Liu, Sagar Mittal, Prathyusha

Aragonda, Vera Lychagina, Younghee Kwon, Michael Wong at Google. «Tenzing

A SQL Implementation On The MapReduce Framework»

[9] Jeffrey Dean and Sanjay Ghemawat at Google. « MapReduce: Simplified Data Processing on Large Clusters».

[10] Martina-Cezara Albutiu, Alfons Kemper, Thomas Neumann. «Massively Parallel Sort-Merge Joins in Main Memory Multi-Core Database Systems».

[11] Dengfeng Gao, Jose Alvin G. Gendrano, Bongki Moon, Richard T. Snodgrass, Minseok Park, Bruce C. Huang, Jim M. Rodrigue. «Main Memory-Based Algorithms for Efﬁcient Parallel Aggregation for Temporal Databases».

[12] Werner Vogels. «Eventually Consistent».

[13] Daniel J. Abadi, Samuel R. Madden, Nabil Hachem. «Column-Stores vs. Row-Stores: How Different Are They Really?».

[14] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach

Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber at Google. «Bigtable: A Distributed Storage System for Structured Data».

|  |  |
| --- | --- |
|  |  |