The Design of Multidimensional Data Model Using Principles of the Anchor Data Modeling: An Assessment of Experimental Approach Based on Query Execution Performance

RADEK NĚMEC, FRANTIŠEK ZAPLETAL
Department of Systems Engineering
Faculty of Economics, VŠB - Technical University of Ostrava
Sokolská třída 33, 701 21 Ostrava
CZECH REPUBLIC
radek.nemec@vsb.cz, frantisek.zapletal@vsb.cz

Abstract: - The decision making processes need to reflect changes in the business world in a multidimensional way. This includes also similar way of viewing the data for carrying out key decisions that ensure competitiveness of the business. In this paper we focus on the Business Intelligence system as a main toolset that helps in carrying out complex decisions and which requires multidimensional view of data for this purpose. We propose a novel experimental approach to the design a multidimensional data model that uses principles of the anchor modeling technique. The proposed approach is expected to bring several benefits like better query execution performance, better support for temporal querying and several others. We provide assessment of this approach mainly from the query execution performance perspective in this paper. The emphasis is placed on the assessment of this technique as a potential innovative approach for the field of the data warehousing with some implicit principles that could make the process of the design, implementation and maintenance of the data warehouse more effective. The query performance testing was performed in the row-oriented database environment using a sample of 10 star queries executed in the environment of 10 sample multidimensional data models. The results show comparison of differences between results of query execution in the environment of the experimental "Anchor" schema and the traditional "Star" schema using statistical methods. The results show possible indications towards expected benefits of the proposed approach that embraces high level of normalization of the resulting database schema in contrast with the traditional approach that results mostly to the creation of a non-normalized database schema.

Key-Words: - multidimensional view of data, multidimensional data model, experimental approach, Anchor modeling, query execution performance

1 Introduction

The usage of information and communication technologies (ICT) gained its firm place in the everyday life of many companies. Numerous empirical analyses document the positive impact of ICT on economic growth, productivity, usefulness and efficiency [1]. One of key aspects of the globalization is that it made sophisticated information technologies affordable for a vast number of companies in the business environment. This trend led to an increase in demand for specialized solutions, allowing analysis of huge amounts of data and reporting of trends. These tasks are main purpose of existence of a specialized software category, commonly known as Business Intelligence (BI). The BI is however mainly an umbrella term for various tools, technologies, architectures, processes, databases and methodologies. These aspects enable effective management and decision-making through high quality information and application of specialized software tools [2].

After years of relative attenuation of the development in the field of BI, the global economic crisis unveiled new topics of discussion among practitioners and researchers. Although the potential of BI itself was clearly recognized, 2 main issues arose and are still discussed. How can the idea behind the application of the BI be extended and how to ensure the BI projects to become more successful, cheaper and eventually even more suitable for small and medium companies?

The first issue is quite successfully addressed by applications from the field of Competitive Intelligence which is recognized as a successor to the BI [3]. These tools tend to use new sources of information to enhance potential of traditional BI tools. The second issue is however addressed by more fields. These fields share a core idea – make the process agile and more user-centric. Therefore new disciplines like agile project management, agile data modeling, agile data warehousing started to appear. Both fields build up much of the whole research effort that is currently related to the BI and its application using modern information technologies and information sources in company's decision-making process. Both fields also share the same effort to incorporate also unstructured content (big data) into the decision-making process to add relevance with most recent events in the society and in the market. This paper focuses on presenting an assessment of an innovative concept, as a part of research activities concerning the second issue mentioned.

1.1 Success of the information system through the quality and relevance of design

BI tools are intended to supply key business users with information that they actually need. Information output should be in proper structure and should be available on time – information with such parameters is vital to gain actionable business insights [4]. The process of building company's BI system should then embrace steps that assure high quality information outputs with highest value for the business user. The paradigm in which the BI system is used primarily on the executive level of company's management has changed. The new paradigm involves usage of BI tools even on tactical and operational level of management. Also novel methodologies that focus on building the BI system in shorter time, with lower costs, while addressing important high priority requirements are present in the current paradigm. Today, the field of industry where the BI could be implemented in is no longer a limiting factor. The current driver of BI implementation is its value for the business (despite size of a company's) which made the BI even more pervasive.

The new paradigm also unveiled several gaps in traditional methodologies which made room for innovation in approaches that are already standardized in the industry. Also, quicker pace of business put more emphasis on the management of requirements, since they can change literally overnight. BI project's stakeholders should be informed how well are project's success dimensions performing and if their own expectations of project's success correspond with the actual state of the project [5]. A methodology should then embrace methods that cover solving of these issues.

With respect to these facts, latest development in the design and development of information systems (including the BI) already confirmed benefits of planning, developing and implementing key parts of information system incrementally. The "agile movement" enforces incremental fashion of the design and development process and offers a set of interesting ideas and principles that deal with issues of execution of time pressured projects. However, the agile should not be treated as a mere synonym of faster design and development of the software [6]. It should also be treated as a philosophy of the whole process of planning and executing actions in the process of company's development over time. This includes also management and implementation of changes in the company and execution of tactical plans and operational tasks. Their outcomes can then be mirrored as changes in data models (supporting company's information system) and these changes should be therefore carefully managed.

In this paper we deal with one aspect of information system's design – the development of a multidimensional data model for a BI system's data storage. We try to present and assess a proposal of an innovative approach to the design of such data model. Our approach should be applicable in the agile oriented process of the design of BI system's data model. Since the BI system should reflect changes in business processes and requirements related to them, there are certain issues that should be covered at the conceptual level of design of the multidimensional data model. The conceptual level of the data model should be as flexible as possible to allow seamless adaptation to changes which occur very often. Current state of conceptual modeling is that the model is often created continuously and concurrent with data loading, accessing and other database management activities [7]. The idea of our approach fits into this presumption: it should offer adaptability of the data model on one side with unique and flexible way of handling changes in key data values and on the other side with minimum specific requirements in the field of data loading.

1.2 The aspect of time in the design of the BI system's data storage

The agile orientation of the BI system's data storage design process is important to ensure the system to meet current requirements and presents valuable and actionable outputs. This fact is also mentioned by Tumbas and Matković [8] who add that the agile orientation in development of computer systems also allows users to change their requirements more frequently without serious consequences. One of information system's success dimensions is timeliness and currency of information [9]. These aspects are very important since relevance of the decision-making process depends mostly on the timeliness and accuracy of available information outputs[1].

The data, as a source of business critical information, often comprise history of changes in values of key business entities (brand names, customer names, organization unit numbers and names, department assignment shifts etc.). These changes are also emphasized as typical changes in the field of data warehousing [10]. It is therefore imperative to facilitate effective means of capturing these changes and incorporate them in the decision-making process while maintaining adaptability of the underlying data model (preferably on the conceptual level of design). The quality of underlying data is a natural and very important antecedent of information quality [11]. Also, an indirect influence on system's quality was proved. These general success dimensions with other mentioned aspects complete the picture of prerequisites of overall information system's success.

Our approach includes conceptual means of capturing changes in attributes' values since this is an important step in the process of BI system's data storage design. Ignoring tracking of changes in business dimensions (used to focus measurement of performance in a business processes) is mentioned as a critical mistake in the process of design of BI system's data storage [12].

Also, an assumption on the high performance of the BI system is an important issue that the project team has to deal with before deploying the system. This issue is also addressed in the assessment of our proposed approach as this issue is commonly mentioned as one of typical aspects of the system quality [13].

1.3 The aim and structure of the paper

The aim of the paper is to present proposal of experimental approach to the design of the multidimensional data model. The paper will also present assessment of the proposed approach from a query execution performance perspective. In the assessment we use sample set of SQL queries and we execute them in the environment of 10 sample multidimensional data models. The results will show whether the proposed approach offers any benefits in terms of better query execution performance along with particular benefits for the usage of the OLAP oriented data storage.

The rest of the paper is organized into 5 sections. Section 2 presents description of the proposed approach and section 3 presents the methodology of the evaluation of the proposed approach from the query performance perspective. In section 4 we present analysis and discussion of results of the evaluation. Section 6 concludes results and presents overview of further research in the topic of the paper.

2 The Proposal of the Experimental Design Concept

The proposed concept is intended to be a counterpart to the traditional approach which is generally based on the construction of a relational multidimensional data model, typically with a star or a snowflake topology [27], [28]. Both approaches, i.e. the proposed one and the traditional one are based on the interpretation of steps of the dimensional modeling by Ralph Kimball [12]. This modeling technique consists of a set of steps that result into the selection of a relevant set of facts at a desired level of detail and analytical viewpoints to allow multipurpose analysis of the facts. Analytical viewpoints (i.e. dimensions) represent entities of the reality that can be used for analytical purposes. Dimensions with their descriptive properties (attributes) and facts form a *multidimensional view of data*. The multidimensional view of data represents the desired way of how the typical BI system's user thinks when analysing performance of respective business processes. The aspect of time is another important aspect of the multidimensional view of data either for the dissemination of changes in key data values but also for the time related analysis of facts. Proper level of detail of facts (the *granularity* or *grain*) influences usability of the data model by BI system's users and therefore it is one of the factors that determine quality of the data model. Facts are used for analytical and further for planning purposes as measures of business process performance. All these aspects of multidimensionality are naturally relevant for our proposed approach since the way how BI system's users think during the analysis of business process performance doesn't change.

Although the dimensional modeling allows selecting proper set of dimensions with further determination of their contents (attributes that

[1] These aspects are relevant to both historical and real-time event processing oriented data sources and information derived from them.

provide descriptive context to facts with implicit hierarchy), the process of the design of multidimensional data model can differ significantly.

The application of the dimensional modeling technique usually results is the definition of the structure and expected contents of a business process related *multidimensional model.* The multidimensional model is usually a semantic-conceptual description and/or visualisation of the multidimensional view of data. Every subset of the multidimensional model is related to specific aspects of respective business process or sub-process in which there is the desire to establish or improve current state of decision making and performance analysis. The multidimensional model is incrementally amplified with the knowledge of newly acquired or revision of current business requirements which can be an uneasy task. The multidimensional model is then incrementally transformed into the *multidimensional data model*. The multidimensional data model can take a form of a relational schema (with star or possibly a snowflake topology) or a schema of classes and their relationships (the object-oriented approach to modeling the multidimensional view of data). This is the logical level of modeling the data model of the multidimensional view of data. The transformation into the physically implementable form is then a natural step in terms of testing and further use of such data model. The proposed approach supports all these mentioned aspects but it differs in the way how the evolution of the database schema is treated. The main difference is in the way how the logical level of the design is performed. The logical relational representation of the multidimensional view of data (i.e. relational multidimensional data model) is component-based and it follows selected principles of the anchor data modeling technique.

2.1 Short description of basic design principles of the anchor data modeling technique

In this paper, we focus on a relational data modeling approach, called the *anchor data modeling*. The anchor modeling is a database modeling technique that facilitates agile development of a database schema [26]. It is formerly focused on the development of the data model of a data warehouse according to the Inmon's approach [17], even if the resulting database schema is not normalized into the 3rd normal form. However, authors do not specify the usage of the technique solely for the purpose of building enterprise data warehouse nor whether their approach is inappropriate or impossible to use for the design of the multidimensional data model.

Anchor modeling is based on a finite set of constructors and principles that are understandable and easy to implement in any relational database environment. The authors specify that their approach should bring several benefits. In context of the design of the multidimensional data model, there is namely the implicit possibility to develop the data model iteratively and incrementally with easier and more effective temporal querying, absence of null values, and reusability of schema objects with related aspects of storage efficiency. There is also better query execution performance expected which should be supported by the existence of a query optimizer's functionality called the elimination of tables in a joins. However, this functionality is not fully implemented in every available database management system as reported in [26].

The database schema resulting from the application of the anchor modeling technique is an *anchor schema*. The anchor schema is a database schema which consists of a finite set of anchors, ties, attributes and eventually knots. Abstract visualization of these components and their representation as a logical relational data model are depicted in figure 1.

The anchor schema is a highly decomposed database schema which is characterized by high level of normalization. The anchor database schema satisfies fully the 5th normal form but the relation can also exist in the 6th normal form which is an extension of the 5th normal form. The 5th normal form is generally based on the decomposition of relations into further irreducible components (mini-relations) that cannot be further decomposed without losing any information contained in it. An Attribute or a Tie relation in the anchor schema satisfies the assumptions of the 6th normal form if they contain additional temporal validity indication attribute [29]. The temporal validity attribute is however facultative and therefore the anchor schema can also partially satisfy the assumptions of the 6th normal form.

The time validity attribute contains information on the time at which a specific value of the attribute started and analogously stopped to be valid with regard to the evolution of the entire entity's state to which the attribute logically belongs. The need for the evidence of changes is usually emphasized as one of key features of a data warehouse (or generally the data storage of the BI system) [20]. Mere overwrite of the value is not an optimal solution in the BI system. Otherwise the system would be inflexible and it would lack future potential to absorb both horizontal (data values) and vertical changes (structure).

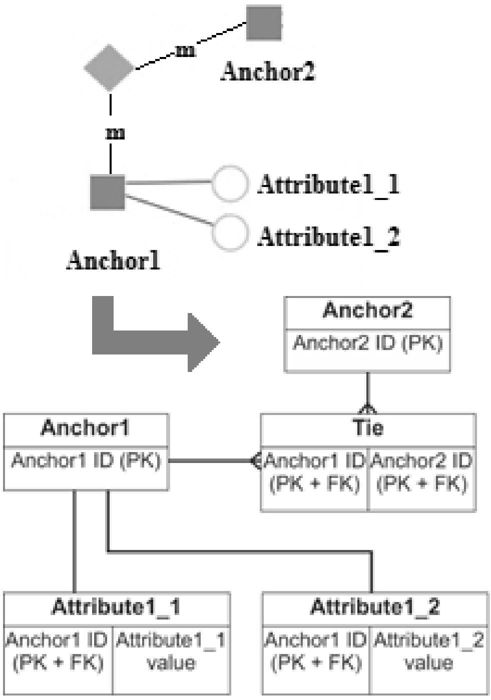

Fig. 1. Basic conceptual constructors of the anchor modeling technique (top) transformed into their logical relational representation (bottom) - “PK” means primary key and “FK” means foreign key

The 6th normal form then allows distinguishing changes in attribute’s values over time on the value level of resolution. This could be beneficial in comparison to traditional approaches in which sometimes whole *n*-tuple must be repeated or a special history-tracking-relation should be used to store changes in a specific attribute or a whole set of attributes. These approaches are commonly known as Slowly Changing Dimension (SCD) and Rapidly Changing Dimensions (RCD) algorithms [12]. The anchor modeling should be especially beneficial in case of temporal querying which is tightly related to the evolution of data values in time [26]. However the main benefits are expected in the field of managing RCD’s for such there are standardized techniques including e.g. splitting the dimension into 2 parts – one part changes sometimes and the other one changes frequently (forming a mini-dimension).

The *Anchor* represents common entity (product, customer, employee etc.). Logical relational representation: a relational table *A*(K#), with 1 column K where K is a primary key of *A*.

The *Attribute* represents a property of an anchor (entity). Logical relational representation: a relational table *Attr*(K*, P), typically with 2 columns where K* is a primary key of *Attr* and a non-null foreign key to respective anchor *A*(K#) at once (a composite key). The domain of P is any non-null data type.

Attribute can be historized, static, knotted static or knotted historized, according to the respective combination of other concepts of schema enrichment. With respect to the design of the multidimensional data model, the historization is especially interesting as it generally means addition of a column which holds the information on temporal validity of values. The relation *Attr* will then be extended to *Attr*(K*, P, T) where the domain of T is a non-null time (or date and time) data type and primary key of *Attr* is then a combination (K*, T).

The *Tie* represents association between two and more entities (anchors) and it is an implicit many-to-many relationship constructor. Logical relational representation: relational table *Tie*($K*_1$,…, $K*_n$), where *n* means total amount of associated *Anchors*, and each K_i for i = {1,…, n} is a foreign key to respective *i*-th *Anchor*. Primary key of the *Tie* is a subset of K_i for i = {1,…, m} where *m* means total amount of *Anchors* that are mandatory to be a part of the primary key of the *Tie* (thus uniquely identifying each tuple of the *Tie* relation). With regard to the design of the multidimensional data model there is an implicit assumption that all related dimensions’ primary keys should be used to uniquely identify each fact. We therefore assume that n=m and *m* will be equal to the total amount of dimensions in the dimensional model.

Knot components and knotted *Attributes* and *Ties* were not used in our sample models because we initially wanted to maintain certain degree of simplicity of the resulting multidimensional data model. Therefore this constructor will not be explained, but respective detailed information on the usage of the *Knot* is contained in [26].

2.2 The proposal of the approach to the design of the multidimensional data model using principles of the anchor modeling technique

In this paper we propose approach to the modeling of the multidimensional view of data based on selected principles of the anchor modeling technique. We apply several alterations according to our previous research since we see it as an interesting alternative to traditional approach where the star schema commonly represents the multidimensional data model with some common

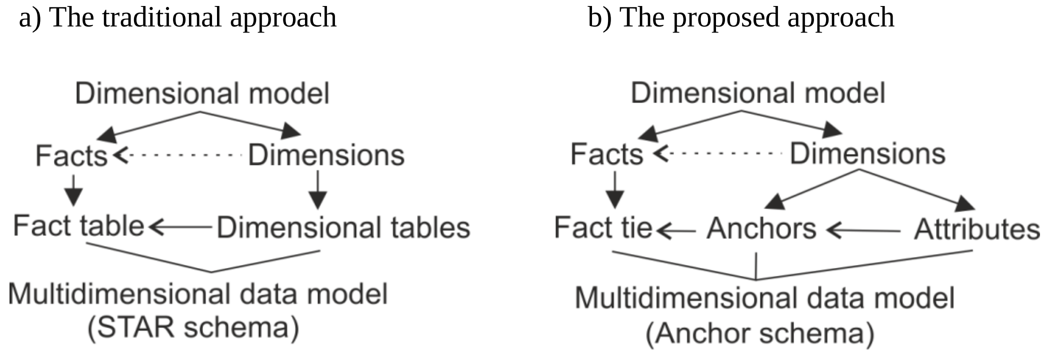

Fig. 2. The flow of approaches to the design of the multidimensional data model

drawbacks. The resulting data model is built using typical relational modeling principles. Therefore it should be applicable in any existing BI system architecture without severe investments into new technologies or upgrades.

The following specification describes the differences in application of the proposed approach and the traditional approach – the figure 2 illustrates briefly the flow of steps in both approaches.

In the traditional approach, each dimension is logically represented as a relation with *r* attributes (including 1 surrogate primary key). The multidimensional database schema then contains at least one fact relation (table) and *n* dimensional relations (tables), each providing context to specified facts. Each dimensional relation is usually non-normalized. Facts are included as attributes in the central fact table which contains {1,…, *n*} foreign keys as realizations of relationships between facts and related dimensions. Relationships between dimension and fact tables have a star topology and in some case also the snowflake topology.

In our proposed approach, dimensions are decomposed into *n* logical *Anchor* objects A_n^{DIM}, where *n* is the total amount of dimensions and a set of *m* related *Attribute* objects $Attr_{n,m}^{DIM}$, where *m* = {1,…, *r*-1} and *r* is a complete formal set of attributes of the respective dimension. Our approach doesn't need surrogate primary keys for $Attr_{n,m}^{DIM}$as they are already contained in each A_n^{DIM} and therefore there can be *r*-1 *Attribute* objects in the schema. Each *Attribute* object $Attr_{n,m}^{DIM}$ is related to 1 respective *Anchor* object A_n^{DIM} using composite key, related to the primary key of the *n*-th A_n^{DIM} (through the foreign key).

Logical relational representation of facts is the modified variant of the *Tie* relation (Tie_{FACT}), which adds also measures into the *Tie* relation. The principles implied by anchor modeling authors suggest leaving all *Attributes* related only with a respective *Anchor* and no other than composite or mixed key should be present in the *Tie*. That would however result in a construction of a special part of the anchor schema with 1 another *Anchor* and several related *Attributes* that would represent only measures (facts). This approach was observed as suboptimal in terms of query execution performance in [30]. The alternative is the Tie_{FACT} relation that has a similar structure as the fact table of the traditional approach: Tie_{FACT}(K*$_1$,…, K*$_n$, M$_1$,…, M$_j$) where M is set of j measures. K is a set of *n* parts of the composite key of the Tie_{FACT} relation, i.e. parts of the primary key and foreign keys pointing to respective *Anchors* A_n^{DIM} at the same time. The resulting database schema is also the *anchor schema*, but with a finite amount of A_n^{DIM}, $Attr_{n,m}^{DIM}$ and Tie_{FACT} objects.

Figures 3 and 4 in the appendix show examples of the multidimensional data model for the dimensional model M4 (later used in the assessment of the proposed approach). Figure 3 shows the traditional variant, i.e. the non-normalized star schema. The figure 4 shows the same model but constructed as an anchor schema using also principles of our approach (the Fact Tie has no special appearance yet).

We presume that several tangible benefits can be related with application of the proposed approach in terms of modeling the data model of the multidimensional view of data:

1) the usage of the historization allows handling various types of changes in dimensions in a unified and flexible way – an alternative to classical methods and algorithms for handling Slowly and Rapidly Changing Dimensions,
2) seamless extensibility and alterability of the schema, e.g. without the need to break or restructure the whole dimensional relation

as in the traditional approach, also with implicit ability to exclude null values) and even with possibility to solving some big data related issues (temporally and structurally evolving data structures used for multidimensional analysis) supports the iterative (agile) orientation of the BI system's database design process,

3) better query execution performance additionally supported also by the possibility of ordering and/or compressing each dimensional property (attribute, i.e. *Attribute* in the anchor schema) separately.

These topics illustrate either expected application but also research potential of the proposed approach and subsequent topics to which we wish to focus our further research. The conceptual constructors that our approach uses are not only understandable with little initial learning effort but also allow their almost instant translation to the database schema of the multidimensional data model. The proposed approach can be used as a tool for the support of an agile oriented design method or methodology for the multidimensional data model design process. The usability of the proposed approach starts on the semantic/conceptual level of design. It can be used to visualize expected form of the multidimensional view of data and its components (enhancement in visualization capabilities are still missing) and thus enhancing communication with the users. The resulting database schema then exhibits modularity characteristics which allow to alter the resulting schema with less effort or to operatively implement extensions of the multidimensional data model into the form of a database schema, according to new or altered requirements. All these aspects are able to fill the gaps in the analysis and design process where the designers of the BI system's data model struggle with the lack of tools to communicate and collaborate with users more effectively and to create the model in a shorter time period. The ability to transform the conceptual multidimensional model into the physically implementable schema more effectively is one of key expected advantages of the proposed approach. The enhancement in semantic and conceptual expressivity of the anchor model is however one of concerns that we also wish to address in our next research.

Uniqueness of handling the RCD's are also potentially beneficial to situations where the rapidly changing data on customers are used very frequently in analytical reports. Indeed there is more effort to be made to maintain additional mini-dimensions that are commonly used to solve existence of rapidly changing properties of the RCD's. Our proposed approach uses unified historization method which is however applicable not only for Attributes but also for Fact Ties although methods similar to eventual historization of facts are already known. The application of the historization for Attributes can however surpass the use of mini-dimensions and establish more effective means of handling RCD's.

The potential of the proposed approach goes further. The topology of the anchor schema is similar to the way how the columnar data store stores columns of the relation. The usage of columnar data stores often contributes to the quality of the BI system and usually makes the querying more powerful. However it is another database solution that should be managed. The implicit modularity of the Anchor schema allows compressing each Attribute separately which is also more powerful when the data is ordered in some way. All these mentioned facts go along with lower hard disk input/output demands that are also mentioned in the section 5. Besides this the separate compression and ordering of each Attribute is easily achievable with our solution and it therefore offers similar potential as the usage of the columnar data store. The main difference is that the current relational database solution doesn't have to be changed. Column-oriented optimization features are already reported in the newest 2012 version of the Microsoft SQL Server and we wish to make further comparison also using this new feature and obtain query performance testing results.

3 The Methodology of the Evaluation of the Proposed Experimental Approach

3.1 Specification of the test setup and sample multidimensional data models

For the purpose of assessment of the proposed approach we used 10 common dimensional models that are typical in various business situations, according to Ralph Kimball [12]. The table 6 in the appendix of the paper contains the overview of all 10 models. The number of facts is low since we wanted to test the initial concept on a smaller simulated dataset but with relevant structure and contents. Therefore dimensions have varying amount of attributes and also number of members (rows). The reason of selection of such sample is that we wanted to include a wider sample of typical situations and to prepare the ground for further research. In the future we would like to focus on the possible application of the proposed approach in a specific business environment (given by the focus of each dimensional model). The 10 selected multidimensional models were then transformed into 2 forms, the traditional non-normalized schema ("Star") and the highly normalized schema ("Anchor"), both variants of a relational database schema. Schemas were implemented into the environment of the Microsoft SQL Server 2008 R2 database server with following hardware configuration: 2x CPU Intel XEON E5450 3GHz, 16 GB RAM. The observed total size of the Anchor schema was however higher than the schema of the traditional variant. The lowest differences were detected for dimensions of models that have a few attributes (about 50 to 80 MB). For bigger dimensions there was difference from about 100 to 300 MB detected, depending on the total amount of dimensions and attributes in them. Nevertheless even the highest difference is not very high also given the fact that the capacity of today's data storages takes values of petabytes. The difference is a result of the primary key values' multiplication in each Attribute relation.

Real instances of multidimensional data models were unavailable during testing. Therefore the content of each model was partly created using machine generation of data (mostly values of facts) and partly using sets of sample typical values, like surnames, department names, product names, order states etc. The generation of the content was controlled so that there are no dimensions' members (rows) that do not correspond to any fact. Also the inclusion of correct values in dimensions was checked so that we will be able to design working queries for testing purposes.

3.2 Methodology of the query execution performance testing

For each dimensional model (and subsequently multidimensional data model) there were 10 unique business questions designed (i.e. total of 100 unique business questions). Business questions were adapted from, or inspired by the set of similar questions that are used in the TPC-DS benchmark (a performance benchmark suite for decision support systems[2]). Each question simulates one situation in which a BI system's user manipulates the interface to get the desired information.

Queries on a multidimensional dataset compute projections, or views, of the underlying data cube [31]. Nevertheless the projections in the SQL statement take into account usually only a small portion of the total amount of dimensional attributes and a smaller amount of rows due to the application of filtering predicates. Business questions were then translated into such queries using Structured Query Language (SQL), with 2 versions for each business question (one for the Star schema and one for the Anchor schema). The set of queries contained samples of queries that involved either all members of specific dimension (mostly small dimension) or a relevant selected portion of it (larger dimensions). Also the number of fact rows involved was not the same. We used aggregation of performance measures (facts) by a relevant measure of time (e.g. year, quarter etc.) or a business relevant category (e.g. product department, customer's home city etc.).

When testing the execution performance of a query, in theory, a test run does not fail if all requests produce correct answers and the state of the test database is unchanged after the execution of the whole test run [32]. Therefore we monitored also the percentage of query processing errors during test runs. All results exhibited 0 % of errors so all test runs were considered successful.

Each query was executed 300 times to avoid significant distortion of results by eventual outlying values. The open-source software Apache jMeter was used for the purpose of testing and recording results of query execution along with the Microsoft SQL Server Management Studio 2008 software.

4 The Analysis and Discussion of Results with the Outlook of the Future Research

The data we gathered from the query execution testing includes 2 types of data sets for both variants of the schema. The first data set contains 300 execution time results for each of the 100 queries and all 10 models (in milliseconds). These results were then used to complete the second data set that contained related characteristics for each query, that is, besides mean value of execution time of each query, also amount of joins required by each query, resulting number of rows returned after the execution of each query, total size of the output (in bytes) and also cardinality of each data object that was scanned by the database system to get the desired output.

First we tested whether the difference of query execution times achieved for each schema variant is statistically significant at the $\alpha = 0.05$ level of significance. Due to the general idea of testing the

[2] http://www.tpc.org/tpcds/default.asp

query performance of the Anchor schema variant in contrast with the traditional Star schema variant (as an initial state of each multidimensional model before the conversion to the Anchor schema variant), the query execution results should be treated as dependent samples. Therefore we used the Student's paired samples statistical t-test to test the significance of the difference between execution performances of each of the 100 pairs of queries.

In the first part of the analysis we were generally interested in finding whether there is a statistically significant positive or eventually zero mean difference of differences $d_{i,j} = X_{i,j,1} - X_{i,j,2}$. Value $X_{i,j,1}$ represents the i-th execution time result for j-th query (i = 1,…, 100; j = 1,…,300) for the traditional Star schema variant (indexed as 1) and $X_{i,j,2}$ represents the analogous execution time result for the Anchor schema variant (indexed as 2). The resulting mean execution time for each query and schema variant is calculated according to the following formulas:

$$\mu_{i,1} = \frac{1}{N}\sum_{j=1}^{N} X_{i,j,1}$$

$$\mu_{i,2} = \frac{1}{N}\sum_{j=1}^{N} X_{i,j,2} \qquad (N=300;\ i=1,\dots,100)$$

In the paired samples t-test's null hypothesis, we assume that the difference of means of execution times of each query equals 0, i.e. H_0: $\mu_{i,1} - \mu_{i,2} = 0$) and the alternative hypothesis is then H_A: $\mu_{i,1} - \mu_{i,2} \neq 0$. According to the resulting difference of means, we can then report also the average magnitude of the difference between query performance results for each query. We used two-tailed significance testing at the same significance level as stated earlier in the text, i.e. $\alpha = 0.05$.

The positive difference then tells, on average, that the Anchor schema variant performs better than the traditional schema counterpart. The reason why we decided to use the mean is that the statistical significance of mean differences can be quite easily tested by numerous statistical tests. Also thanks to the large number of observations for each query (N=300) the results in such data set are usually less noticeably affected by outlier values and the statistical tests' results are less prone to the non-normality of the distribution of values in the sample.

We also applied the Student's paired samples t-test to determine whether the means of both sets of each query's execution time means are significantly different (H_0: $\mu_1 - \mu_2 = 0$, i.e. both variant's result sets ("Star" and "Anchor") are equal with the alternative hypothesis H_A: $\mu_1 - \mu_2 \neq 0$). Results of paired samples t-tests for each query are in the table 1.

As for the initial paired t-test result the difference of means of both variant's query execution results is on average -64.02 ms (standard deviation of differences is 330.14 ms). This result indicates a bit faster query execution results in favor of the Star variant but the value is statistically non-significant with p>0.05 (p=0.055). Therefore we can reject the null hypothesis with 5 % chance of error and the difference is then rather given by a chance variation. This fact is also supported by the value of the Pearson coefficient of correlation between query execution time results of both schema variants,

Table 1. Query execution time differences between Anchor and Star schema variants (in milliseconds, positive difference indicates better performance of the Anchor schema variant)

Query # / Model	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
M1	-43^{**}	-605^{**}	-114^{**}	-55^{**}	96^{**}	-72^{**}	161^{**}	-4	-1132^{**}	97^{**}
M2	140^{**}	539^{**}	133^{**}	97^{**}	9^{**}	132^{**}	91^{**}	216^{**}	9^{**}	-6^{**}
M3	3	292^{**}	16^{**}	13^{**}	-3	-53^{**}	-166^{**}	-427^{**}	-441^{**}	7^{**}
M4	24^{**}	32^{**}	3^{**}	0	-100^{**}	-3^{**}	-816^{**}	-5^{**}	3^{*}	-25^{**}
M5	-7^{**}	-11^{**}	3	-9^{**}	4	1	23^{**}	-5	169^{**}	2
M6	63^{*}	187^{**}	-93^{**}	-1499^{**}	-2^{*}	-74^{**}	-349^{**}	3	3^{*}	222^{**}
M7	7	-81^{**}	-81^{**}	-363^{**}	-92^{**}	-64^{**}	-19^{**}	-362^{**}	260^{**}	-163^{**}
M8	-82^{**}	-1738^{**}	-69^{**}	-36^{**}	-59^{**}	-23^{**}	-35^{**}	325^{**}	-4	-33^{**}
M9	-5	113^{**}	-134^{**}	596^{**}	73^{**}	57^{**}	55^{**}	641^{**}	-104^{**}	-502^{**}
M10	-2^{**}	-36^{**}	-382^{**}	-682^{**}	-178^{**}	251^{**}	115^{**}	39^{**}	-285^{**}	8^{**}

N=300; * difference is significant at the 0.05 level (p<0.05); ** difference is significant at the 0.01 level (p<0.01), non-significant results are highlighted with grey color which is $r_{xy} = 0.971$ (*p*=0.00). Although, on average, both results sets are not significantly different there are interesting results on the query level of resolution. According to the contents of the table 1 there are total 38 queries for which the Anchor schema variant performed better and 49 queries for which the Star schema variant performed better. Besides that, there are other 13 queries for which the *t*-test indicated statistically non-significant results. However, when looking at the resulting differences there are many values closer or very close to zero among the significant and non-significant results. This is about 27 results if we consider values from 0 to about 10 for example. However, a more precise evaluation of low values in terms of the upper negligibility threshold can be carried out with knowledge of a time value which would determine that a certain difference between duration of 2 events is indistinguishable for the human eye.

In the second part of the analysis of our approach we were interested in the assessment of further relationships between the query execution time and accompanying output aggregated results of physical processing characteristics of each query. First we determined whether there is a significant correlation between mean execution time of queries and the amount of joins required to get the desired output of each query. It is clear that our approach brings higher possible amount of joins due to the nature of the resulting multidimensional data model's structure (the traditional approach needs lower amount of joins). Therefore there is a concern that the more dimensional attributes (i.e. $Attr^{DIM}$) are included in the query the more time it should take to process such query. However, in the case of a standard business question, there will be some filtering predicate mostly present and the projection will include only a small subset of all dimensional attributes, as it was in our case. The computed Pearson coefficient of correlation between the amount of joins and the mean execution time is $r_{xy} = -0.084$ (non-sig.) for the Anchor schema variant and $r_{xy} = -0.009$ (non-sig.) for the Star schema variant. This means that the mean execution time has no linear relationship with the amount of joins required to answer the business question as it can be also seen in tables 2 and 3 but there is naturally some correlation between other specific summary results and the mean execution time (*avg*).

The correlation with the total size of the output (*size*) is very high for both variants and correlation with rows returned as output (*rows*) is significant at the 5 % level but not very high. This is due to the fact that even lesser amount of rows returned can be a product of a more costly query.

Table 2. Correlation between *avg* and other output - results for the Star schema variant (*N*=100)

	Avg	**Rows**	**Size**	**Joins**
Avg	1			
Rows	0.636**	1		
Size	0.945**	0.658**	1	
Joins	-0.009	0.066	0.046	1

** Correlation is significant at the 0.01 level (2-tailed)

Table 3. Correlation between *avg* and other output - results for the Anchor schema variant (*N*=100)

	Avg	**Rows**	**Size**	**Joins**
Avg	1			
Rows	0.615**	1		
Size	0.922**	0.658**	1	
Joins	-0.084	0.155	0.031	1

** Correlation is significant at the 0.01 level (2-tailed)

Table 4. Results related to the relational table scanning operation (*N*=100)

	Anchor	**Star**
Mean rows scanned	27702	23992
Mean I/O demands	0.1316	0.3186
Correlation with Avg		
Mean rows scanned	0.156	0.093
Mean I/O demands	0.155	0.175

A possible issue related to our proposed approach is the number of dimension's members (rows) that are to be scanned when executing the query (according to the execution plan of the query). The assumption is that the more dimensional attributes are used in the query (and more joins are therefore required in our approach) the more cardinality should be associated with primary key index scan operation. The table 4 contains summary results related to explanations regarding the aforementioned assumptions. As can be seen in the table, the mean count of rows scanned by the query optimizer in the schema constructed according to our approach is higher than the mean for the traditional approach. However the total input/output (I/O) demands are lower 2.4 times for our approach than for the traditional approach. The coefficient of the correlation between the total rows scanned and total I/O is non-significant and very low. Besides the fact that there is no linear relationship with the mean execution time (probably also due to the lower amount of queries tested) we see that our normalized approach has positive effects on the total input/output demand although the mean rows scanned is higher. Although the results of the paired samples *t*-test do not indicate that our approach is more beneficial than the traditional one at this point, but as for the I/O intensity, our approach has less demands even with the higher amount of required joins.

There are 3 queries where the difference between mean execution times is higher than 1 second. After analysing the execution plan of each of the queries there are several notable findings. The important indicator is also the total cost of the execution plan which is a dimensionless value used for the evaluation of the best execution plan to be used for the query processing task. The query M8Q2 returns 345 rows and execution plans look very similar to each other. However there is an aggregation operation used differently in the Anchor variant that seems to demand a lot of system resources, although all other operations are similarly I/O intensive in both variants. The aggregation is however performed on facts and therefore there is probably a problem related with an error in calculations due to outdated table statistics of the fact table/fact tie table maintained by the database system. Also the M6Q4 exhibits the same problem with table statistics since the total cost projection of the Anchor variant's execution plan is even lower and there are no notably more I/O intensive processing operations. Also the query performance results for the query M1Q9 appears to suffer from inconsistencies in storage statistics since there are also no notable differences in I/O demand of operations and the execution plan also has lower processing costs indicator. Other differences lower than 1 second but higher than half a second are caused by a larger count of rows returned as output due to rather imprecise definition of filtering predicates and usage of more than 3 dimensional attributes as output. This caused higher I/O demands of the scanning operation.

According to the results there are signs of the fact that the high degree of normalization which is related with our proposed approach exhibits no severe issues regarding average execution time of tested star queries (given by the resulting correlation coefficient and also the paired samples *t*-test). We will conduct more research on finding more proofs of soundness and further usefulness of our proposed approach which appears to be perspective.

5 Related Works

There are several works dealing with modelling aspects of the multidimensional data model design process more thoroughly. However no works dealing with the application of the anchor modelling or any similar approach that is based on high level of the normalization in this field were found. In [26], there is the anchor modeling technique introduced but as for testing the querying performance the authors did performance tests of their basic concept only in comparison with results for the centralized enterprise data warehouse schema alternative (a schema normalized into the 3rd normal form). This approach is however formally related to the Inmon's "Corporate Information Factory" approach [14]. This approach was already thoroughly researched and assessed in the past that lead to a recommendation of a tighter relationship with R. Kimball's pure data mart and multidimensionality oriented approach [12]. The resulting hybrid approach and the latter pure approach respectively are the background for our approach and also for several other works cited. The main motive for such combination of approaches is that the CIF approach based BI system's data model is not suitable for ad-hoc querying [15]. The strong point of the CIF oriented data model however lies in the field of data mining and standardized production reporting. On the other side the Kimball's approach is capable of satisfying needs of the ad-hoc analysis of large amounts of data. The usual trouble is with supporting general tasks that need more normalized data structures (i.e. sophisticated data analysis tasks).

Papers [20], [22], [23], [24] and [25] follow principles of Kimball's approach and they deal with aspects of possible extension of the standard E/R modeling paradigm for the purpose of modeling the multidimensional view of data. In [10], there is a proposal and discussion of a relational schema transformation concept which facilitates adaptation of the data warehouse to changing data sources. The changes in either structure or contents of the multidimensional database schema is however still a viable research topic. In [20], there is an approach called MultiDimER proposed that is also a visual conceptual modelling language. However the MultiDimER includes interesting dimensional attributes' hierarchy handling concepts that take into account also evolution of the hierarchy levels and dimension's content. The MultiDimER approach uses hierarchical decomposition of attributes with respect to possible change behaviour. The approach is however not practically assessed in the paper. The proposal is however advanced enough to be practically applicable with additional and appropriate definition of implementation specifications. We plan to enrich our approach with visual conceptual tools for representing multilevel hierarchies although we believe that a rigid formal definition of a hierarchy is not as important topic since the advent of self-service BI tools. [22] also proposes conceptual design concepts that are targeted at the thorough formalization of a snowflake schema. Our approach is in fact a special form of the snowflake schema which is in a more normalized star schema. The authors present only formal assessment of their approach and the paper lacks practical example of their proposed approach. Also means for handling hierarchies are missing. [23] and [24] propose an advanced conceptual design concept that takes into account not only formal design of dimensions and facts but also varying forms of hierarchies. The paper however lacks practical evaluation of the proposed approach and despite formal richness of the conceptual model definition (including visual modeling constructors), it gives only theoretical justification of the concept. The justification does not present enough proofs whether such solution would be easily implementable without serious knowledge and understanding of the concept by designers but mainly by users. State-of-art of our proposed concept is based on simple and understandable representation of constructors for the multidimensional data model components with compliant logical and also conceptual meaning. Further extensions however should not sacrifice understandability for extended formal richness. In [25], the proposal focuses also on the conceptual modeling aspects including advanced formal definition of modeling constructors. Nevertheless no hierarchy modeling aspects are included as in [23] and [24] and also the concept lacks practical evaluation – the approach proposed in this paper however appears to be quite universal and extensible.

Our approach can also be used for the conceptual level of modeling with unified set of simply understandable constructors. However, useful concepts of effective handling changes in and visualization of hierarchies are still missing – only changes in attributes are already covered by the historization principle. Despite that we provided initial evaluation from the query performance perspective which is important for the evaluation of possible conceptual modeling capabilities and usefulness as well. However despite that, our results should be treated as experimental at this point with more research to be conducted to prove its soundness and usefulness in practice. The results however show some viable potential for possible practical use of the proposed approach and also possible further research topics were already outlined.

Mentioned papers present different research track opposite to another one that should be also noted. This research track enforces usage of object oriented data modelling principles and UML language to model structural and also temporal evolution aspects of the multidimensional data model. Papers [16], [17], [18], [19] present novel approaches that introduce interesting conceptual design principles for modelling the multidimensional data model. However these works present no exact performance testing results besides examples of possible usage of their approach and formal requirements evaluation. Another concern with these proposed approaches is that the object oriented approach is sill not proved to be successful or beneficial enough in the field of supporting the multidimensional data model design and implementetion process. Paper [21] also follows the UML based research track but the authors propose an approach that adds solution for handling changes of data values in the object-oriented multidimensional data model. This approach is similar to our proposed approach in the aspect of handling the changes in dimensional attributes (using objects of special subclasses that handle the changes). Nevertheless our approach is not intended as part of the object oriented research track.

Table 5 summarizes cited research works with shorter description of approaches discussed in them.

Table 5 Summary of other approaches to the design of multidimensional data model

Author(s)	**Characteristics of approach**
Malinowski, Zimányi (2006) [20]	Relational, multidimensional data model, with the solution to handle changes of data values (MultiDimER conceptual model, a visual modeling language proposal with temporal evidence capabilities)
Ravat, Teste, Zurfluh (1999) [21]	Object-oriented, multidimensional data model, UML based, with solution to handle changes of data values (use of special classes for historization of instances of dimension’s members)
Trujillo, Palomar,	Object-oriented, multidimensional

Gomez, Song (2001) [18]	data model, UML based approach to conceptual design of the data warehouse's data model, without the solution to handle changes of data values
Abelló, Samos, Saltor (2002) [16]	Object-oriented, multidimensional data model, UML based, without the solution to handle changes of data values, special semantic object-oriented relationships between multiple star schemas
Nguyen, Tjoa, Wagner (2000) [17]	Object-oriented, multidimensional data model, UML based, without the solution to handle changes of data values, extensions to handle OLAP operations
Gosain, Nagpal, Sabharwal (2011) [19]	Object-oriented, multidimensional data model, UML based, without the solution to handle changes of data values, focus on dimension hierarchies
Levene, Loizou (2003) [22]	Relational, multidimensional data model, without the solution to handle changes of data values, special interest in the snowflake schema usage (decomposition of the multidimensional data model)
Regardt, Rönnbäck, Bergholtz, Johannesson, Wohed (2009 [26]	Relational, data model of the enterprise data warehouse, with the concept of historization to handle changes of data values
Sapia, Blaschka, Höfling, Dinter (1998) [25]	Relational, multidimensional data model (conceptual extensions to standard E/R modeling to handle multidimensional data), without the solution to handle changes of data values
Kamble, Franconi (2004) [23] and Kamble (2008) [24]	Relational, multidimensional data model (conceptual extensions to standard E/R modeling to handle multidimensional data), without the solution to handle changes of data values

5.1 The summary of the outlook for the further research

The actual state of our research is that we are dealing only with performance differences results of standard star queries. In our current research we are dealing with finding more information on the anchor database schema's behaviour and explanations of main differences between query execution plans. Further we would like to study benefits of the historization concept in context of commonly known SCD and RCD algorithms. Another future course of research is the usage of the conceptual representation of the anchor model for the support of requirements analysis process regarding the more effective multidimensional data model design (agile orientation of the analysis and design process).

We will also analyse effects of Attribute relation compression and different ordering strategies to the query performance as opposed to the execution of the same star queries in the column oriented data storage environment (current results were obtained in the row oriented data storage environment as stated before). The columnar data stores compress attributes of the table in a similar fashion like it is also possible using our approach. The implementation of our approach in the column store is possible nevertheless we expect problems regarding the negative multiplication effect of implicit modularity of the schema and compression. The real differences are however yet to be tested but we first intend to assess column-oriented features in traditional relational database systems.

6 Conclusion

The aim of the paper was to present proposal of the experimental approach to the design of the multidimensional data model. The approach was assessed from a query execution time perspective. The results show that as for differences between query execution time results of both assessed database schema variants, our sample of queries produced statistically insignificantly different results although the mean for both samples was by 64 ms worse for the Anchor schema variant the results. The results also showed that there was unequal amount of better and worse performing queries for the Anchor schema variant. The differences were however not high in most cases which is the logical consequence of statistically insignificant difference in results (given by the results of the Student's paired samples *t*-test). The database schema created according to our proposed approach exhibits high level of normalization which has some expected benefits (higher query execution performance is among them) and also possible drawbacks. Required higher amount of joins proved to be insignificantly correlated with the mean query execution time. Also the total amount of rows processed during execution of the scan operation (selection of appropriate dimensional table's rows) proved to have low correlation (non-sig.) with the mean query execution time. Differences higher than 1 second were indicated as a result of outdated table statistics because there were no notable changes in query execution plan structures (the total plan cost was mostly even lower for the Anchor schema variant). The results of total I/O demands also indicated that our approach brings lower demands in terms of required amount of disk operations despite the fact that the amount of rows scanned is in some cases higher (even double).

There may be other effects that can influence query execution results especially in case of extremely complex queries. We plan to conduct more research on these situations as well as aspects of historization that can serve as a more effective alternative to common SCD and mainly RCD algorithms. The concept of our approach surely needs more proofs of possible usefulness although some of them were already outlined in the text. These future results will help us to fully justify applicability of the proposed approach in terms of its efficiency, scalability and they will also possibly present complex proofs of its soundness for the field of the multidimensional data model design process.

7 Acknowledgement

This paper was made with financial support of Student Grant Competition research project SP2012/184 “The analysis of data warehouse’s database schema modeling characteristics with a focus on agile approach to Business Intelligence system development” and the European Social Fund within the project CZ.1.07/2.3.00/20.0296.

References:

[1] P. Doucek, J. Fisher and O. Novotný, Impact of ICT on National Economies - Open Issues, *Proceedings of the 20th International Information Management Talks IDIMT-2012*, 2012, pp. 111-114.

[2] E. Turban, R. Sharda and D. Delen, *Decision Support and Business Intelligence Systems*, 9th ed., Prentice Hall, 2010.

[3] C. Murphy, *Competitive Intelligence: Gathering, Analysing And Putting It to Work*, Gower Publishing Ltd., 2005.

[4] B. Wixom and H. Watson, The BI-Based organization, *International Journal of Business Intelligence Research*, Vol. 1, No. 1, 2010, pp. 13-28.

[5] C. Barclay, Towards an integrated measurement of IS project performance: The project performance scorecard, *Information Systems Frontiers*, Vol. 10, 2008, p. 331–345.

[6] S. Ambler, *Agile Database Techniques: Effective Strategies for the Agile Software Developer*, Wiley, 2003.

[7] N. Roussopoulos and D. Karagiannis, Conceptual Modeling: Past, Present and the Continuum of the Future, *Conceptual Modeling: Foundations and Applications. Lecture Notes in Computer Science*, Vol. 5600, 2009, pp. 139-152.

[8] P. Tumbas and P. Matković, Agile vs Traditional Methodologies in Developing Information Systems, *Management Information Systems*, Vol. 1, 2006, pp. 15-24.

[9] W. H. DeLone and E. R. McLean, Measuring Success: Applying the DeLone & McLean Information Systems Success Model, *International Journal of Electronic Commerce,* Vol. 9, No. 1, 2004, pp. 31-47.

[10] E. A. Rundensteiner, A. Koeller and X. Zhang, Maintaining Data Warehouses over Changing Information Sources, *Communications of the ACM*, Vol. 43, No. 6, 2000, pp. 57-62.

[11] R. R. Nelson, P. A. Todd and B. H. Wixom, Antecedents of Information and System Quality: An Empirical Examination Within the Context of Data Warehousing, *Journal of Management Information Systems*, Vol. 21, No. 4, 2005, pp. 199-235.

[12] R. Kimball a M. Ross, *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*, 3rd ed., Wiley, 2013.

[13] W. H. DeLone a E. R. McLean, The DeLone and McLean Model of Information System Success: A Ten-year Update, Vol. 19, No. 4, 2003, pp. 9-30.

[14] W. H. Inmon, *Building the data warehouse*, 4th ed., Wiley, 2005.

[15] A. R. Patel and J. M. Patel, Data Modeling Techniques for Data Warehouse, *International Journal of Multidisciplinary Research,* Vol. 2, No. 2, 2012, pp. 240-246.

[16] A. Abelló, J. Samos and F. Saltor, YAM2 (Yet another multidimensional model): An extenxion of UML, *Proceedings of the International Symposium on Database Engineering & Applications*, 2002, pp. 172-181.

[17] T. B. Nguyen, A. M. Tjoa and R. Wagner, An object-oriented multidimensional data model for OLAP, *Proceedings of the International*

Conference on Web-Age Information Management, 2000, pp. 69-82.

[18] J. Trujillo, M. Palomar, J. Gomez and Song, Designing Data Warehouses with OO Conceptual Models, *IEEE Computer, Special Issue on Data Warehouses,* Vol. 34, 2001, pp. 66-75.

[19] A. Gosain, S. Nagpal and S. Sabharwal, Quality Metrics for Conceptual Models for Data Warehouse focusing on Dimension Hierarchies, *ACM SIGSOFT Software Engineering Notes,* Vol. 36, No. 4, 2011, pp. 1-5.

[20] E. Malinowski and E. Zimányi, A conceptual solution for representing time in data warehouse dimensions, *Proceedings of the 3rd Asia-Pacific conference on Conceptual modelling,* 2006, pp. 45-54.

[21] F. Ravat, O. Teste and G. Zurfluh, Towards Data Warehouse Design, *Proceedings of the eighth international conference on Information and knowledge management*, 1999, pp. 359-366.

[22] M. Levene and G. Loizou, Why is the snowflake schema a good data warehouse design?, *Information Systems,* Vol. 28, No. 3, 2003, pp. 225-240.

[23] A. Kamble and E. Franconi, A data warehouse conceptual data model, *Proceedings of the SSDBM*, 2004, pp. 435-436.

[24] A. S. Kamble, Conceptual Model for Multidimensional Data, *Proceedings of the 5th Asia-Pacific Conference on Conceptual Modelling (APCCM 2008)*, 2008, pp. 29-38.

[25] C. Sapia, M. Blaschka, G. Höfling and B. Dinter, Extending the E/R model for the multidimensional paradigm, *Proceedings of the ER Workshop on Data Warehousing and Data Mining*, 1998.

[26] O. Regardt, L. Rönnbäck, M. Bergholtz, P. Johannesson and P. Wohed, Anchor Modeling: An Agile Modeling Technique Using the Sixth Normal Form for Structurally and Temporally Evolving Data, *ER 2009,* vol. 5829, No. 1, 2009, p. 234–250.

[27] D. Riazati and J. A. Thom, Matching Star Schemas, *DEXA 2011, Part II, LNCS 6861*, 2011, pp. 428-438.

[28] S. Rizzi, Conceptual Modeling Solutions for the Data Warehouse, *Data Warehouses and OLAP : Concepts, Architectures and Solutions*, IGI Global, 2007, pp. 1-26.

[29] C. J. Date, H. Darwen and N. A. Lorentzos, *Temporal Data and the Relational Model: A Detailed Investigation into the Application of Interval and Relation Theory to the Problem of Temporal Database Management*, Elsevier, 2003.

[30] R. Němec, The Comparison of Anchor and Star Schema from a Query Performance Perspective, *World Academy of Science, Engineering and Technology,* Vol. 71, 2012, pp. 1718-1722.

[31] E. Torlak, Scalable Test Data Generation from Multidimensional Models, *Proceedings of the SIGSOFT/FSE'12*, 2012, pp. 1-11.

[32] A. Askarunisa, P. Prameela and N. Ramraj, DBGEN- Database (Test) GENerator - An Automated Framework for Database Application Testing, *International Journal of Database Theory and Application*, Vol. 2, No. 3, 2009, pp. 27-54.

Appendix

Table 6. Overview of 10 multidimensional data models used for the assessment of our proposed approach

Model	Specification of the model content			
	Dimension	**Attributes**	**Size (rows)**	**Fact table content and size (rows)**
M1	Month	7	156	HR snapshot (7170000)
	Employee	7	517	
	Store	9	1000	
M2	Date	13	4749	Store inventory snapshot (7000000)
	Product	11	10000	
	Store	9	1000	
M3	Month	7	156	Monthly account balance snapshot (7000000)
	Household	9	200000	
	Account	10	220000	
	Account state	3	4	
	Banking product	4	20	
	Branch	4	1000	
M4	Date	13	4749	Order transactions (6684500)
	Customer	7	30000	
	Sales representative	6	517	
	Product	11	10000	
	Order status	3	5	
	Transport type	3	5	
	Payment type	3	3	
M5	Date	13	4749	Web sales channel profitability (7000000)
	Daytime	4	86400	
	Product	11	10000	
	Promotion	12	509	
	Visitor	9	50000	
	Channel	5	21	
M6	Date	13	4749	Project defects (7000000)
	Project	5	10000	
	Project status	3	4	
	Project priority	3	5	
	Client	6	8000	
M7	Date	13	4749	College students attendance events (7000000)
	Student	10	96000	
	Faculty	6	7	
	Subject	7	1400	
	Room	11	1000	
M8	Date	13	4749	Sales transactions (7000000)
	Store	9	1000	
	Promotion	12	509	
	Product	11	10000	
M9	Date	13	4749	Customer billing snapshot (7000000)
	Line	6	400000	
	Billing plan	6	100	
	Sales representative	7	5000	
	Customer	9	50000	
M10	Date	13	4749	Procurement transactions (7000000)
	Product	11	10000	
	Vendor	7	4000	
	Contract conditions	3	10	
	Procurement transaction type	3	9	

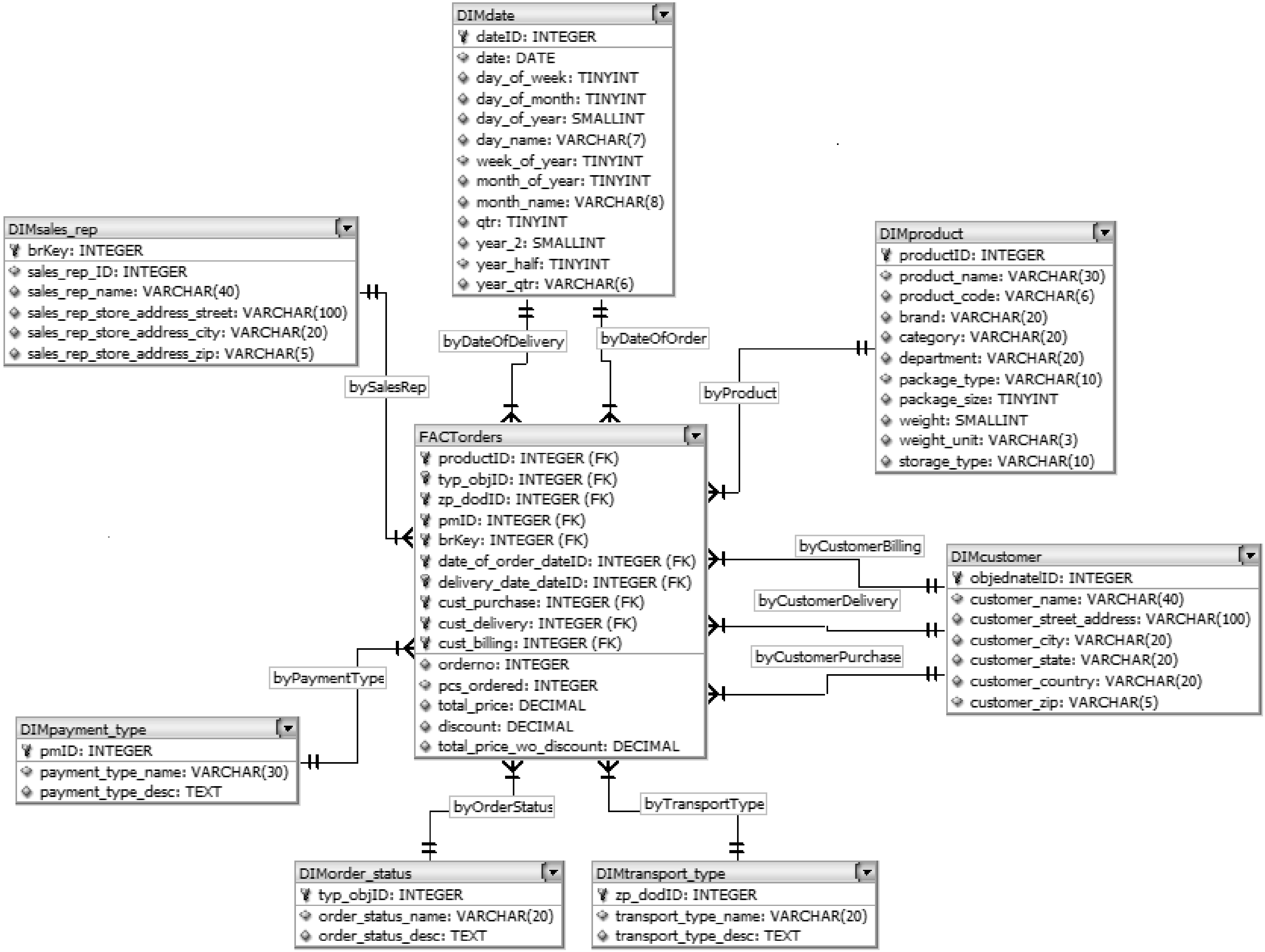

Fig. 3. Example of the model M4, represented as the traditional schema variant ("Star") (logical view)

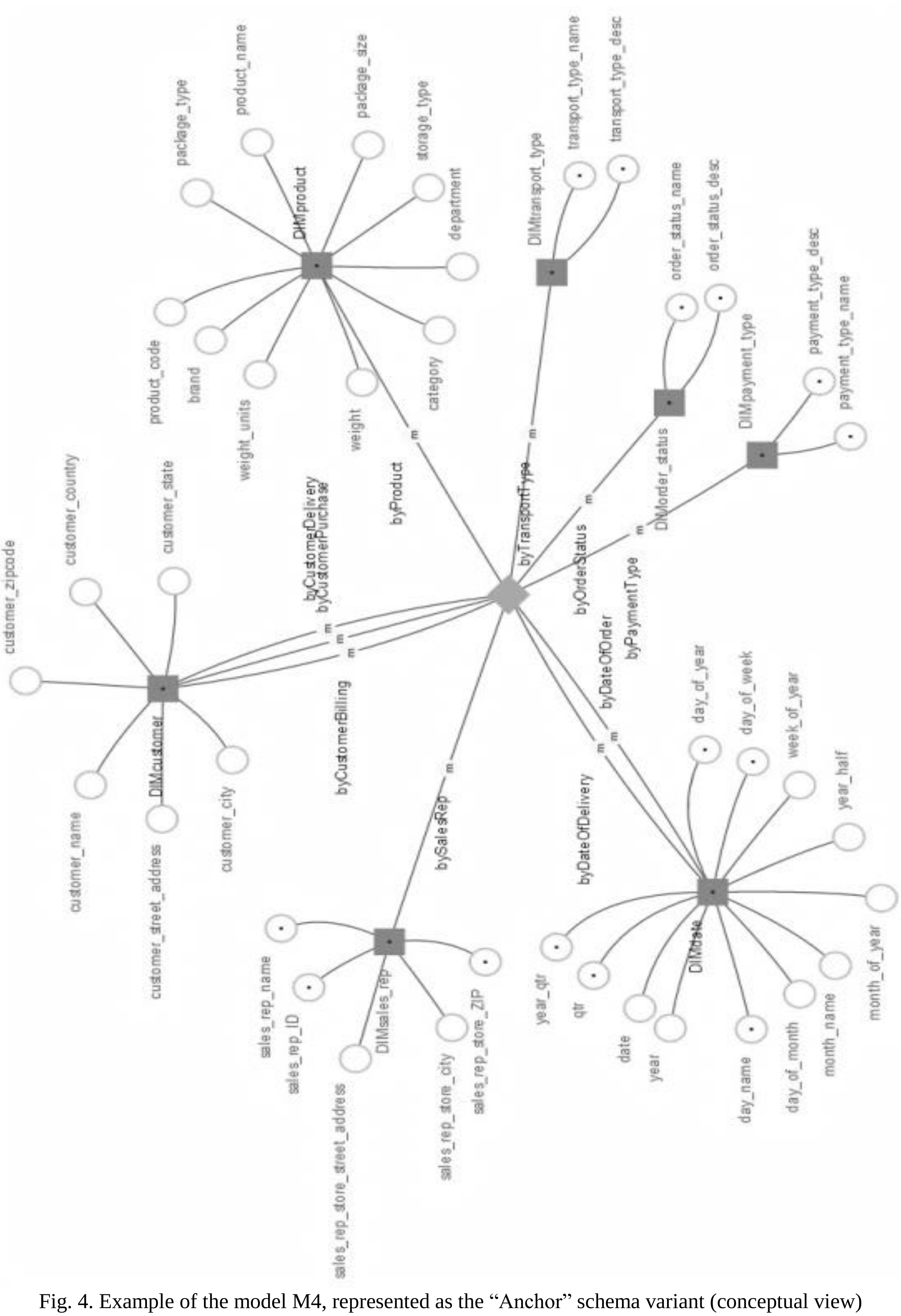

Fig. 4. Example of the model M4, represented as the "Anchor" schema variant (conceptual view)