Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Technology 1 (2012) 107 – 111

Procedia Technology

INSODE 2011

Archiving business operational data heritage

Orlando Belo, Alice Marques

Algoritmi R&D Centre, Universidade do Minho, Portugal

Abstract

Operational Database Systems are keeping large amounts of information that are not used in any aspect on current business processes. It's inactive data that is maintained only for historic reasons or to ensure data patrimony and process heritage. Thus, database archiving is today a critical task for companies that are worried about their data storage devices, processing resources, and, obviously, data preservation. Extracting selectively relevant business information from operational database and storing it on a separate archive data store for consulting it's a good strategy for preserving data as well to reduce storage database resources and improve data processing services. In this paper we propose an archiving technique inspired on the most effective data warehousing dimensional modelling techniques. We designed and developed a set of mechanisms with the ability to analyze and characterize a conventional relational database, and based on a set of business preservation requirements and archiving rules, prepare it to be archived on a multidimensional system designed automatically. We'll present their description and functionalities, describing the way in which they can be used and the manner how operational data is stored in a archiving oriented data warehouse.

Keywords: Database Archiving; Data Warehouses; Archiving Oriented Dimensional Modelling; Multidimensional Structures.

1. Introduction

Nowadays, operational systems frequently became obsolete when compared with the up-to-date database facilities and technological trends in data storage and administration domains. In spite of having old characteristics and low information management versatility they have a huge asset for the organization where they are sited: data. Organizations evolve as well their database management systems, however data remains. Provably, its format and relationships change over time, but the meaning it's the same whenever we need it. Even changing for better and modern database management systems, organizations want to preserve their old systems in a way that they can access and consult whenever they want in future. Often, they require archiving their data patrimony in a different way than a conventional backup, which as we know demands that the old system could be restored [17]. Companies are facing large data patrimonies that are rarely accessed, but need some kind of preservation services. No one likes to loose valuable information even if it is not used currently. Now, it's recognized that there is a life cycle for information, in general, which means that the utility of an information piece begins in a specific moment and ends some time later. Recognizing that, companies are looking for efficient manners to archive the data they are not using now but that can be useful again some day latter [12]. In this paper we propose an archiving technique for conventional operational systems supported by a data warehousing system [6] [10]. Basically, we create a set of mechanisms with the ability to analyse and characterize a relational database in order to prepare it to be archived on a multidimensional system. The idea is preserving data (and metadata) about the most important business information entities, relationships, and transactions, in a way that it can be preserved properly and consulted easily when archived through the most basic mechanisms of browsing and reporting that a data warehousing system provide us today.

2212-0173 © 2012 Published by Elsevier Ltd.
doi:10.1016/j.protcy.2012.02.021

2. Related Work

During the last few years, several attempts were made in order to establish automatically dimensional schemas directly from operational systems metadata. The motives to do that are quite diverse and supported by different reasons ranging from the need to generate the most appropriated decision support structures, to the creation of a simple set of views especially oriented for business management. A few years ago, in 2002, Phipps & Davis (2002) developed a specific algorithm that based on a operational schema propose a dimensional schema for a data warehouse, using an extended ME/R model with user refinement requirements. Based on such requirements, the algorithm was also able to validate the generated dimensional schema through a set of queries, adjusting a final schema selection that is closer to such user-requirements. Following this line of working, Golfarelli et al. (2001) [5] developed a semi-automatic technique to build conceptual designs for data warehouses based on XML information sources, a format that they recognised to be very important today in the data warehousing field. Latter, Jensen et al. (2004) [9] presented one of the most relevant works on this area, proposing an automatic process to construct multi-dimensional schemas directly from operational systems metadata. They developed a four-step method – gathering metadata, annotating metadata, relationship definitions, and construction of hierarchies – that disposes a basic structure for an OLAP cube. A different approach was presented by Romero & Abello in 2007 [20] focusing the generation of multidimensional schemas using ontologies, representing specific business domains, combined with a set of reasoning mechanisms special oriented to identify multidimensional patterns. In this same year, Song et al. [21] presented SAMSTARplus, a semi-automatic tool that has the ability to generate the most typical dimensional schemas for data warehouse. Escaping a little from the data-driven approaches, more recently, M. Deshpande [3] presented another automatic process to create dimensional schemas, using *Dimensional Design Patters* [11], and having the ability to identify conformed dimensions as a way to refine generated schemas. In spite of all this interesting works related to the automatic generation of decision-support data structures, some other lines of researching points to other direction: operational systems data archiving. The most basic characteristics of a data warehouse sustain undoubtedly all functional needs of a data archive as well all the services to consult it. A data warehousing design process [7] [8] [14] for archiving it's not very different from a conventional one. Still, we must be careful with the data we want to archive and the its storage format. The rest is very similar.

3. Dimensional Based Archiving Schemas

Data warehousing multidimensional schemas are simple and flexible enough to provide quite efficient means to receive and retrieve archive data and, if necessary, querying data easily, crossing all available data items. Data Warehousing Systems own very adequate and adjustable architectures for archiving solutions, provide the necessary means to gather, transport, treat (when required), receive, store and querying. There are several ways for retrieving the characterization of all entities, relationships, constraints (domain, systems rules, enterprise restrictions, etc.), and, of course, attribute functional dependencies [1] [13] [16]. However, this particular type of schema does not follow the same orientation lines that we use to apply in the development of a conventional decision-support data structure for a data warehouse. Although most of the design directives can be applied with success, since data structures have the same configurations and combination (join) paths. But, their purposes are quite different. In the case of archiving, we are worried to preserve the most relevant entities and their involvement with other database objects that were (in some aspect) important to support business activities. Basically, we want to preserve the business history (and its facts) of a particular company. Database archiving is not a simple process. We need to know what, and how to archive. As referred in [17] we don't archive a complete database system, but only some selective business records that are inactive for a specific time slice.

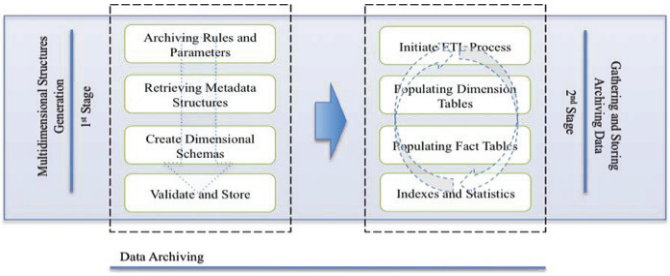


Fig.1 – Archiving stages and processes

In the development of our model, we assumed the existence of some preliminary archiving rules, reflecting the company's archiving policy for regulating the process. This is not new. It's typical from any company that have to manage archives daily [19]. We preferred to have a global view of the entire company's data patrimony and ensure a robust archiving system to maintain company's business data heritage. The processes that materialized our archiving model (Fig.1) are executed into two different stages: 1) *Multidimensional structures generation* – that integrates a preliminary process set for data archiving setup and configuration, including dimensional structures generation and validation: it's here where we configured the archiving strategy and data warehousing dimensional data structures; and 2) *Gathering and storing archiving data* – that includes all data manipulation processes from information retrieving to data storing in the data warehouse; all these processes are responsible to populate the data warehouse, bringing data from operational system to the data warehouse. Stage one is executed once, basically because it integrates configuration system tasks. However, second stage's processes are executed every time an archiving order is launched.

```
PROCEDURE MultidimensionalStructuresGeneration(Period)
        /** Archiving Rules and Parameters
(s01)   archivingParametersDefinition(Period,ParametersList).
(s02)   transactionLogAnalysis(Period,ObjectList).
(s03)   userQueryingAnalysis(Period,PreferenceList).
(s04)   archivingRulesGeneration(Period,ObjectPreferenceList).
        /** Retrieving Metadata Structures
(s05)   retrievingDatabaseSchema(DataBaseId,Schema).
(s06)   factTablesIdentification(Schema,FactTables).
(s07)   dimensionTablesDefinition(Schema, FactTables, Dimensions).
        /** Creating Dimensional Schemas
(s08)   creatingArchivingDimensionalSchema(FactTables, Dimensions, ArchiveSchema).
        /** Validate and Store
(s09)   dimensionalSchemaValidation(ArchiveSchema,Status, ErrorList).
(s10)   if Status = true
(s11)           then storingSchema(ArchiveSchema).
(s12)           else communicateErrors(ErrorList).
```

Fig.2 – 1st Stage Archiving Steps

4. Archiving Multidimensional Structures Generation

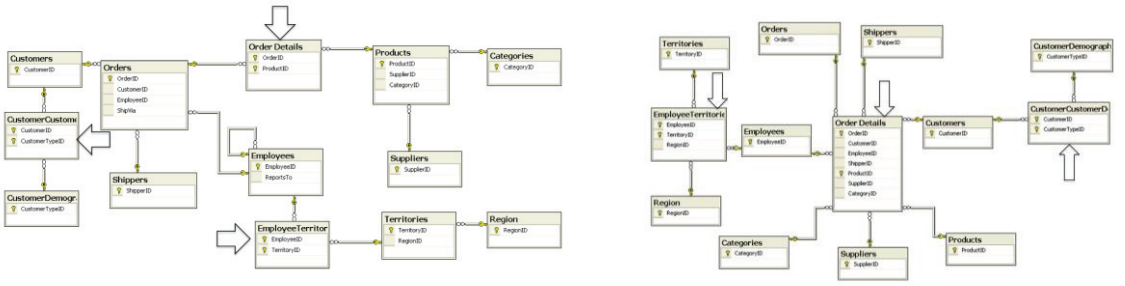
4.1. Archiving Rules and Parameters

We archive data to improve database systems and applications performance, reduce operational costs (processing and storage), reduce supporting infrastructures, or simply to preserve data, just to name a few [4]. Taking into account that 60 to 80 percent of data is not currently used [17], we can contribute a lot to reach such goals promoting a balanced archiving policy for operational systems. The first task of the entire archiving process is one of the simpler. Nevertheless, it regulates the execution of the dimensional schema selection algorithm and, so, the entire archiving process. Archiving rules and configuration parameters are defined only once. For this, we analyse inactive transactions and past querying tasks in order to establish inactive data, and the periods of data availability, data retention, and data refreshment [19], to manage and monitor data archiving populating processes. To accomplish the first part of the process we accessed and analysed systems transactions logs to discover the most candidate tables (especially transaction tables) with significant volume of inactive records, and querying records to evaluate the importance of data for users – assuming that a great number of accesses means important pieces of data. After this semi-automatic task, we ask to the data archive manager the value for each configuration parameter.

With all this information the system generates the archiving rules. Fig.2 presents the sequence of the most important steps (s01-s04) of this task.

4.2. Retrieving Metadata Structures

The generation of metadata structures it was divided into two parts: fact tables identification; and dimension tables definition. Using a physical schema of an operational system, we begin trying to discover potential fact tables. We detected them analysing all transaction tables that exists in the schema provided, recognised them through a simple heuristic, namely: every transaction table having one or more foreign keys but is not itself a reference for other table, potentially is a fact table – it's a similar process than the one presented in [9], but without consider the candidate keys issue because sometimes they appear in transaction tables. The fact table identification algorithm (s06) collects all schema's tables and, for each one, checks the existence of foreign keys accordingly the previous heuristic. If the heuristics will be verified then the table will be identified as a fact table – see tables with pointing arrows in Fig.3 (a).



(a) Initial Operational System Schema (b) Archiving Dimensional Schema

Fig.2 – 1st Stage Archiving Steps

Secondly, the algorithm tries to identify eventual dimension tables. First it gathers all tables for those a fact table has a foreign key. The tables that verified that condition will integrate the first line of dimensions for the referred fact table (s07). The algorithm continues applying the same conditions trying to establish a second line of dimension tables (usually recognized as outriggers), and repeat the process until no more foreign keys will be found. Finishing this, the algorithm collects all the gather metadata structures for fact and dimension tables, and builds automatically the final constellation schema for the archiving dimensional model (s08) (Fig.3 (b)). After this, the schema is presented to the archiving system's manager to validation and error detection (s09). If no errors were detected, the archiving schema is enriched with the rule definitions and configuration parameters metadata (s11). Otherwise, errors are communicated to the archiving system's manager (s12). To test and validate our model and algorithm, we used the 'Northwind' database (http://www.microsoft. com/download/en/details.aspx?displaylang=en&id=23654), a very well know case study example provided by Microsoft for SQL Server database management system.

4.3. Gathering and Storing Archiving Data

Having the archiving schema stable we can configure and execute the process to feed the data warehouse – the 2nd stage of the archiving process -, usually recognised as ETL (Extract, Transform, and Load) process. Basically, we need to follow the source-to-target map developed during the 1st stage. The operations involved are not complicated, a situation that is quite different from a conventional data warehousing populating process. The reason why this happens is due to the fact that data is maintained at row level, not implicating any type of transformation (aggregation, summarization, conversion, quality improvement, etc.). Archiving is simple operation, involving usually a simple copy of the inactive data from a place to another. Thus, having well defined archiving parameters and the source-to-target map implemented (*Initiate ETL Process*) we can execute the populating process every time we need to "refresh" the archive (monthly, quarterly, yearly, etc.). To ensure archive's integrity constraints, we need to feed firstly dimension tables (*Populating Dimension Tables*), from the most peripheral to the ones near fact tables,

and only then the schema's fact tables (*Populating Fact Tables*). Terminating this 2nd stage, we update tables' indexes and archiving statistics.

5. Conclusions

Archiving becomes more and more important to companies each day that passes. They are discovering the real dimension of their data patrimonies and the effects that it causes on daily system operations. In this work we tried to contribute to improve such task providing some means to generate automatically a data warehouse dimensional schema for archiving. As far as we know, this kind of approach is not very usual. Data warehouses were always designed to decision-making support, not for data archiving solutions. In the DBPreserve Project - Data Warehouses for the Long-term Preservation of Institutional Electronic Records and Databases, a project sponsored by FCT (Ref. PTDC/CCI/73166/2006) we developed efforts to prove the adequateness of a data warehousing system for data archiving solutions [2]. It is not a complete new approach. We designed our archiving based data warehousing model and its processes, including the specification of archiving requirements and specificities to be followed during data selection (1st stage) and storing (2nd stage) of the entire archiving process, especially for systems with a great operational longevity. The archiving system optimizes the dimensional schema taking into consideration the use (or not) of the data to be archived, analysing past transaction records and following (when available) user querying paths. Having this information, the system knows what to archive, providing better multidimensional schemas, more adequate for data archiving.

References

1. S. Bell and P. Brockhausen. Discovery of data dependencies in relational databases. In European Conference on Machine Learning, 1995.
2. G. David, Data Warehouses in the Path from Databases to Archives in International Workshop on Database Preservation, 2007-03-23, National e-Science Centre, Edinburgh, Scotland, 2007.
3. M. Deshpande , "Automating Multiple Schema Generation Using Dimensional Design Patterns", MSc Thesis, Department of Computer Science, College of Engineering, University of Cincinnati, May 2009.
4. Forrester Consulting "Why database archiving should be part of your enterprise DBMS strategy", A commissioned study by Forrester Consulting on behalf of Clearpace, January 2008.
5. M. Golfarelli, S. Rizzi, and B. Vrdoljak: Data Warehouse Design from XML Sources. In Proc. of DOLAP, pp. 40-47, 2001.
6. M. Golfarelli, S. Rizzi Data, "Warehouse Design: Modern Principles and Methodologies", McGraw-Hill Osborne Media; 1 edition, May 26, 2009.
7. A. Gutiérrez, A. Marotta, "An Overview of Data Warehouse Design Approaches and Techniques", InCo - Pedeciba, Facultad de Ingeniería, Universidad de la República Montevideo, Uruguay, October 2000.
8. C. Imhoff, N. Galemmo, J. Geiger, "" Mastering Data Warehouse Design – Relational and Dimensional Techniques", Wiley Publishing, Inc., 2003.
9. M. R. Jensen, T. H. Møller, and T. B. Pedersen: Specifying OLAP Cubes On XML Data. JIIS 17(2-3): 255-280, 2001.
10. R. Kimball, M. Ross, W. Thornthwaite, J. Mundy, B. Becker, "The Data Warehouse Lifecycle Toolkit", Wiley, 2nd ed, January, 2008.
11. M.E. Jones and Y. Song, "Dimensional Modeling: Identifying, Classifying & Applying Pattern" Proceedings of the 8th ACM International Workshop on Data Warehousing and OLAP, DOLAP 2005, Bremen, Germany, pp. 29–38.
12. J. Lee, "Database Archiving: A Critical Component of Information Lifecycle Management" in
Database Journal, April, 2004.
13. S. Lopes, J. Petit, L. Lakhal, Efficient Discovery of Functional Dependencies and Armstrong Relations. In Proceedings of the 7th International Conference on Extending Database Technology: Advances in Database Technology (EDBT '00), Carlo Zaniolo, Peter C. Lockemann, Marc H. Scholl, and Torsten Grust (Eds.). Springer-Verlag, London, UK, 350-364, 2000.
14. J-N Mazon, J. Trujillo, M. Serrano and M. Piattini, "Designing data warehouses: From business requirement analysis to multidimensional modeling", Proceedings 1st International Workshop on Requirements Engineering for Business Need and IT Alignment, Paris, France, 2005.
15. C. Mullins, "Database Archiving for Long-Term Data Retention", The Data Administration Newsletter, Published in TDAN.com 2006.
16. N. Novelli, R. Cicchetti, FUN: An Efficient Algorithm for Mining Functional and Embedded Dependencies. In Proceedings of the 8th International Conference on Database Theory (ICDT '01), Jan Van den Bussche and Victor Vianu (Eds.). Springer-Verlag, London, UK, 189-203, 2001.
17. J. Olson, Database Archiving Basics, A New Function for Improving Data Management, Information Management Special Reports, September 22, 2009.
18. D. Phipps, K. C. Davis, "Automating Data Warehouse Conceptual Schema Design and Evaluation", In Proceedings of the 4th International Workshop on Design and Management of Data Warehouses (DMDW), Toronto, Canada, 2002.
19. D. Robb, Database Archiving Best Practices: Advice from the Trenches, Enterprise IT Planet: Storage Resources for the Enterprise IT Professional, August 30, 2004. (www.enterpriseitplanet.com/storage/ features/article.php/3401301).
20. O. Romero and A. Abello, "Automating multidimensional design from Ontologies" Proceedings of the ACM tenth international workshop on Data warehousing and OLAP (DOLAP), Lisbon, Portugal, 2007, pp. 1-8.
21. I.Y. Song, R. Khare, B. Dai, "SAMSTAR: a semi-automated lexical method for generating star schemas from an entity-relationship diagram," Proceedings of the ACM tenth international workshop on Data warehousing and OLAP (DOLAP), Lisbon, Portugal, 2007, pp. 9-16.