

12. GI-Fachtagung für Datenbanksysteme in
Business, Technologie und Web (BTW 2007)
5. bis 9. März 2007 - Aachen, Germany
<http://www.btw2007.de/>

Data Warehouse Detective: Schema Design Made Easy

Till Haselmann, Jens Lechtenböcker, Gottfried Vossen*
European Research Center for Information Systems
University of Münster, Germany
haselmann@uni-muenster.de, {lechten,vossen}@wi.uni-muenster.de

1 Introduction

The deployment of data warehouses (DWs), which are integrated databases to support decision making, has become common practice in modern information technology landscapes, and the methodical design process for such databases has been advanced significantly in recent years. In this context, we are investigating issues concerning the quality of DW schemata which can be measured and algorithmically enforced via multidimensional normal forms (MNFs) during conceptual design. So far, however, tool support for DW design based on MNFs has been missing, a gap which we are about to close.

Compared to traditional database design, DW design poses several new challenges arising from the fact that DW design has to take a number of pre-existing data sources into account: On the one hand, DW end-user requirements need to be aligned with the information provided by these data sources. On the other hand, the data sources need to be integrated for analysis purposes based on a multidimensional representation. To address these challenges, we have previously proposed three MNFs [LV03] that formalize (i) correctness and completeness of DW schemata with respect to pre-existing data sources, (ii) avoidance of redundancies, and (iii) context sensitive summarizability in the presence of null values.

2 About the DWD Demo

We are currently developing and field-testing a tool called *Data Warehouse Detective* (DWD) to support the design of normalized data warehouse schemata. A prototypical version of this tool has been designed and implemented in the course of a project seminar involving six IS students during summer term 2006. Our demo for this tool consists of the following three parts.

First, we present how to design a new conceptual DW schema based on an analysis of pre-existing databases. To this end, DWD imports meta-data — i. e., tables, attributes, keys,

*Currently with The University of Waikato Management School, Hamilton, New Zealand

and foreign keys — of selected databases via ODBC. It then enriches these meta-data based on an analysis of database instances to detect missing keys, foreign keys, and functional dependencies, i. e., the meta-data which are often not declared properly in practice. DWD does that by creating and testing hypotheses about candidate constraints. It assumes that a candidate constraint holds if a systematic check does not exhibit any counterexamples. Afterwards, the designer chooses relevant attributes to populate the forthcoming data warehouse and, based on the functional dependencies previously recognized, DWD synthesizes multidimensional conceptual DW schemata for these relevant attributes, including dimension hierarchies, as sketched in [Lec03]. MNFs are used throughout this process to guarantee that all fact schemata under design really fit the pre-existing databases and to gain control over optional dimension levels with NULL values, which allows to avoid summarizability problems and inconsistent queries.

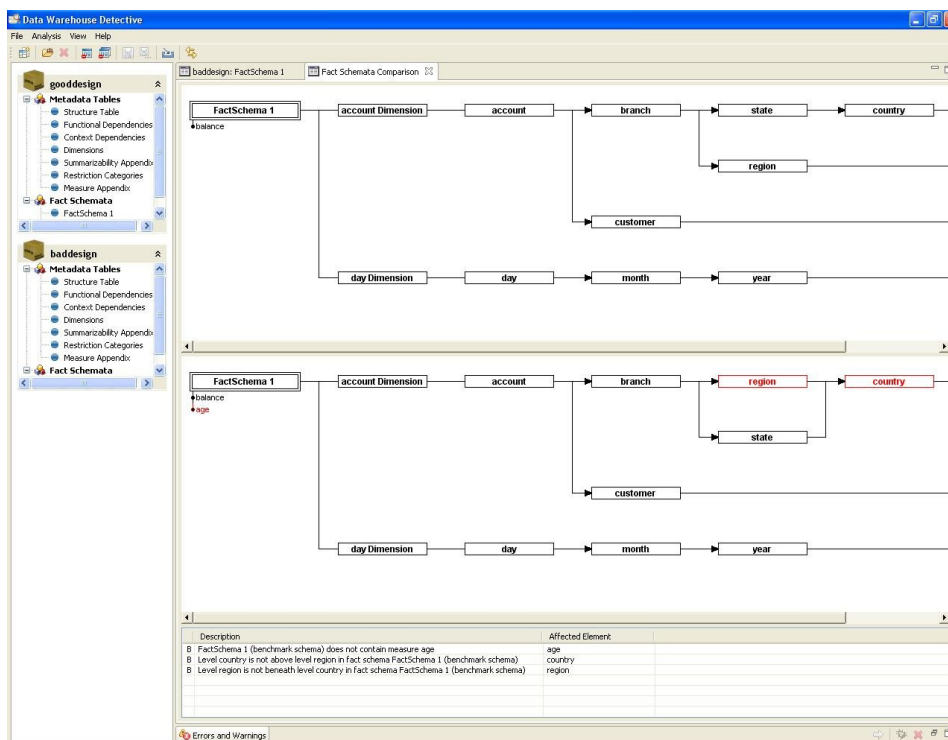


Figure 1: Benchmarking with DWD against MNFs.

Second, we present a benchmarking of existing schemata against multidimensional normal forms, which is illustrated in the screenshot shown in Fig. 1. The lower part of the screen shows a hand-made DW fact schema in the banking domain, while the upper part shows a normalized schema synthesized by DWD based on an analysis of the underlying data sources. Both schemata represent account balances of individual accounts per day along with corresponding dimension hierarchies. Benchmarking these schemata, i. e., comparing

the hand-made schema against the normalized one to detect violations of normal forms, leads to two observations highlighted in red by DWD in the lower schema: First, the hand-made schema is based on the assumption that regions can be assigned uniquely to countries, while the underlying data contradicts this assumption with regions that cross countries, e. g., the Alps. Thus, loading the dimension hierarchies of the lower schema is doomed to fail. Second, the assignment of the age of customers as a measure in an account related fact schema leads to redundancies, as the age of a customer will be repeated for each of his or her accounts. Here, normalization with DWD creates a second fact schema, reporting all customer related measures including age (not shown in the figure).

Finally, we demonstrate how to verify the consistency of DW schemata in the presence of (schema) changes in the pre-existing databases. As extracted and enriched database structures form a major input for DW design with or without DWD, one major challenge throughout DW projects arises from structural changes in data sources. We have started to analyze the effect of schema changes, which often lead to DW schema versions, in [GLRV06]. As an initial step towards analyzing the impact of such changes on DW schemata, DWD allows to check the consistency of a previously modeled DW schema with respect to the current state of data sources. The software reports any mismatches between data sources and DW schemata — e. g., deleted but necessary attributes, new or missing functional dependencies with an impact on multidimensional layout — and allows the user to revise the design accordingly.

As mentioned, DWD is currently under field testing in various DW production environments, where it shows promising performance. More information about DWD is available at <http://www.dw-detective.de/> or directly from the authors.

Acknowledgments

We would like to thank Philipp Borgschulte, Philipp Dopjans, Katja Funke, Markus Heinrich, and Roland Reschka who together with the first author formed the team that did the initial implementation of DWD.

References

- [GLRV06] Matteo Golfarelli, Jens Lechtenbörger, Stefano Rizzi, and Gottfried Vossen. Schema Versioning in Data Warehouses: Enabling Cross-Version Querying via Schema Augmentation. *Data & Knowledge Engineering*, 59(2):435–459, 2006.
- [Lec03] Jens Lechtenbörger. Data Warehouse Schema Design. In *Proc. of the 10th BTW*, pages 513–522, 2003.
- [LV03] Jens Lechtenbörger and Gottfried Vossen. Multidimensional normal forms for data warehouse design. *Inf. Syst.*, 28(5):415–434, 2003.