

Why Data Modeling Is Still Relevant

Written by John Pocknell, Senior Product Manager, Quest



INTRODUCTION

There are those who think data modeling is passé or irrelevant. After all, data modeling theory is more than 30 years old and some data modeling tools have been around for 20 years. But data modeling is still essential. In fact, it may be more necessary now than ever. Let's look at several key reasons why data modeling is experiencing a resurgence, and how effective data modeling can benefit your organization.

The unique value of data modeling

While there are other modeling techniques and notations, such as Business Process Modeling and Unified Modeling Language, data modeling is uniquely designed to accurately capture business data requirements and transform them into a reliable database structural design. The key differentiator is that only data modeling focuses on the "data at rest"; all the other techniques and notations focus on "data in motion."

Another way to say this is that data modeling concentrates on issues that lead to a solid database design, while the other techniques focus on issues that result in better application design or the things useful to programmers, such as data structures, objects, classes, methods, application code generation, and so on.

Case in point: I've personally been an expert witness at several trials in which plaintiffs were suing for serious financial remuneration because custom database applications had performance and/or data accuracy problems. In every single case, the defendant had failed to create the data model for the business requirements, and therefore data effectiveness suffered. In many cases, ad-hoc database design or database design using more programmatic techniques and tools had also resulted in inefficient database design, and no amount of coding trickery could overcome this. So in every case, the plaintiff won.

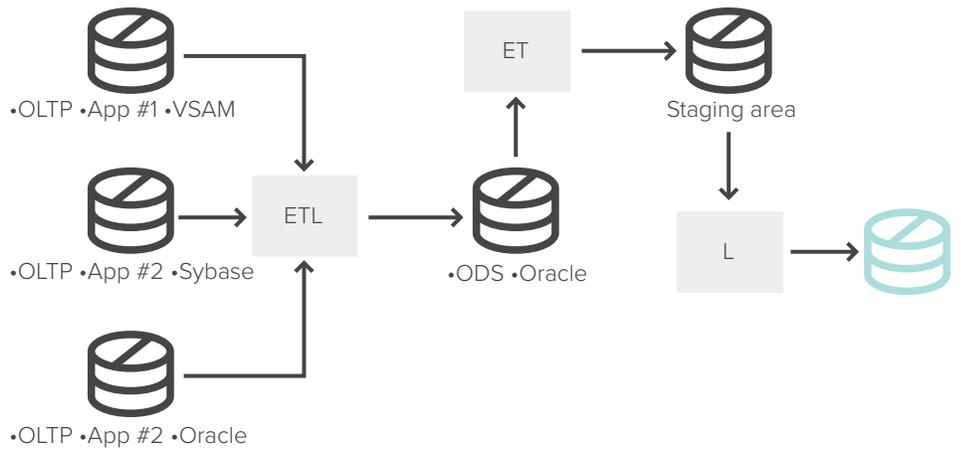


Figure 1. Data warehouse via intermediate ODS

One reason data modeling has seen measurable resurgence is the growth of data warehousing.

Using data modeling in data warehousing

One reason data modeling has seen measurable resurgence is the growth of data warehousing. Because storage is cheap these days, most organizations can afford to retain historical aggregate or summary data for making significant strategic decisions. With the accumulation of numerous source legacy OLTP (online transaction processing) systems, there are two keys ways to approach populating a data warehouse:

- Directly from source to warehouse (see Figure 1)
- Through an intermediary database, which is often referred to as an ODS (operational data store) (see Figure 2)

There is much debate as to which approach is superior; I won't address those arguments here. Instead, I will point out that, regardless of which approach is taken, the database design (i.e., the data at rest) is paramount, because in a data warehouse, the data itself and the business information it contains is the most relevant and valuable asset. Typical data warehouse queries and reports issued via business intelligence tools process that asset to yield key strategic decision-making results.

One key way in which data modeling supports the data warehousing effort is by mapping legacy data fields to their data warehouse counterparts. Carefully mapping front-line business data to the data warehouse helps with the design of queries and reports, as well as with ETL (extract, translate and load) programming

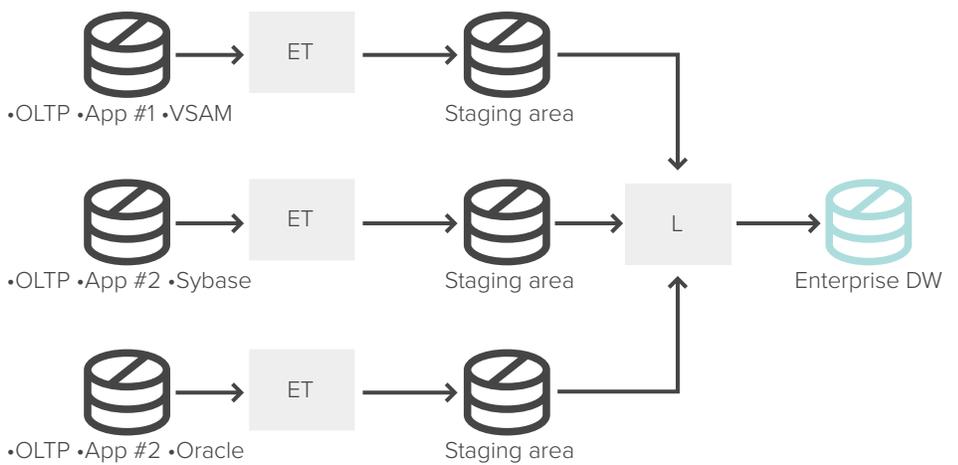


Figure 2. Data warehouse via separate feeds

efforts. Without such a mapping, as your OLTP legacy systems evolve, you would have no automatic tie to the dependent data warehousing information; therefore, you would have to almost totally re-engineer, rather than simply follow, the OLTP source data ramifications that trickle downstream to the DW and BI end points.

Using data modeling in systems development

Data modeling is important not only for data warehousing projects, but also for more traditional OLTP systems development. Unfortunately, people often get so caught up in novel paradigms such as “extreme programming,” “agile software development,” or “scrum” that they compromise on — or even skip entirely — data modeling.

The problem is that these new approaches don’t always spell out exactly how data modeling should be incorporated, and therefore people often forego it. My belief is that no matter what development methodology you choose, data modeling should be integrated into your development process wherever it makes sense.

Figure 3 shows how both conceptual and physical data modeling should fit into an overall database design process — whether it’s for a totally new system or for an existing system that’s being updated or even completely re-engineered.

Data modeling as part of a formal development process

There’s one final reason why data modeling has been getting more attention these days. In many cases, organizations are finally requiring data models as a sign-off deliverable in the development process. I attribute this to organizations attempting to adhere to the Software Engineering Institute’s Capability Maturity Model and Capability Maturity Model Integration concepts. The idea here is quite simple: To make your development process more mature, regardless of technique, you need to think in terms of both the processes and tools used to achieve the desired result. For example, the proper and effective use of project management tools is often cited as the single best way to escalate from Level 1 (Chaos) to Level 2 (Repeatable). The central idea that both processes and tools can lead to maturity has helped to reinvigorate the interest in data modeling.

No matter what development methodology you choose, data modeling should be integrated into your development process wherever it makes sense.

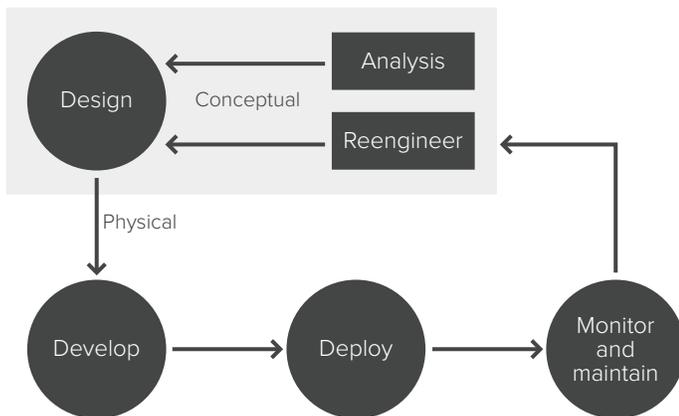


Figure 3. Overview of database design

Data modeling can assist with any effort, regardless of development methodology or paradigm.

CONCLUSION

Data modeling has come a long way since its inception. Even though the heydays of CASE and software engineering ended in the '90s, the need for — and usefulness of — data models has not subsided. Data modeling can assist with any effort, regardless of development methodology or paradigm. So don't pass up data modeling just because it's a mature technique.

ABOUT THE AUTHOR

John Pocknell is a senior manager of product management at Quest. Based in the U.K., John is responsible for the strategy and roadmap for the Toad portfolio of products worldwide. He has been with Quest since 2000, working in the database design, development and deployment product areas, and he has run many Toad training courses for customers.

John has spent the last 14 years successfully evangelizing Toad to customers at various events throughout Europe, the U.S. and the AsiaPac region, and he writes many blogs and papers on the Toad user community, Toad World, as well as technical briefs about Toad on the Quest website. John has worked in IT for 30 years, most of that time in Oracle application design and development.

ABOUT QUEST

Quest helps our customers reduce tedious administration tasks so they can focus on the innovation necessary for their businesses to grow. Quest® solutions are scalable, affordable and simple-to-use, and they deliver unmatched efficiency and productivity. Combined with Quest's invitation to the global community to be a part of its innovation, as well as our firm commitment to ensuring customer satisfaction, Quest will continue to accelerate the delivery of the most comprehensive solutions for Azure cloud management, SaaS, security, workforce mobility and data-driven insight.

© 2016 Quest Software Inc.

ALL RIGHTS RESERVED.

This guide contains proprietary information protected by copyright. The software described in this guide is furnished under a software license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of the applicable agreement. No part of this guide may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of Quest Software Inc.

The information in this document is provided in connection with Quest Software products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Quest Software products. EXCEPT AS SET FORTH IN THE TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, QUEST SOFTWARE ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL QUEST SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF QUEST SOFTWARE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Quest Software makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Quest Software does not make any commitment to update the information contained in this document.

Patents

Quest Software is proud of our advanced technology. Patents and pending patents may apply to this product. For the most current information about applicable patents for this product, please visit our website at www.quest.com/legal.

Trademarks

Quest, Toad and the Quest logo are trademarks and registered trademarks of Quest Software Inc. in the U.S.A. and other countries. For a complete list of Quest Software trademarks, please visit our website at www.quest.com/legal. All other trademarks, servicemarks, registered trademarks, and registered servicemarks are the property of their respective owners.

If you have any questions regarding your potential use of this material, contact:

Quest Software Inc.

Attn: LEGAL Dept
4 Polaris Way
Aliso Viejo, CA 92656

Refer to our Web site (www.quest.com) for regional and international office information.