The Data Warehouse ETL Toolkit

The Data Warehouse ETL Toolkit

Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data

Ralph Kimball
Joe Caserta

Wiley Publishing, Inc.

Published by
Wiley Publishing, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Published simultaneously in Canada

eISBN: 0-764-57923-1

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

Library of Congress Cataloging-in-Publication Data

Kimball, Ralph.
The data warehouse ETL toolkit : practical techniques for extracting, cleaning, conforming, and delivering data / Ralph Kimball, Joe Caserta.
p. cm.
Includes index.
eISBN 0-7645-7923 -1
1. Data warehousing. 2. Database design. I. Caserta, Joe, 1965- II. Title.

QA76.9.D37K53 2004
005.74—dc22 2004016909

Credits

Vice President and Executive Group Publisher:
Richard Swadley

Vice President and Publisher:
Joseph B. Wikert

Executive Editorial Director:
Mary Bednarek

Executive Editor:
Robert Elliot

Editorial Manager:
Kathryn A. Malm

Development Editor:
Adaobi Obi Tulton

Production Editor:
Pamela Hanley

Media Development Specialist:
Travis Silvers

Text Design & Composition:
TechBooks Composition Services

Contents

Acknowledgments

First of all we want to thank the many thousands of readers of the Toolkit series of data warehousing books. We appreciate your wonderful support and encouragement to write a book about data warehouse ETL. We continue to learn from you, the owners and builders of data warehouses.

Both of us are especially indebted to Jim Stagnitto for encouraging Joe to start this book and giving him the confidence to go through with the project. Jim was a virtual third author with major creative contributions to the chapters on data quality and real-time ETL.

Special thanks are also due to Jeff Coster and Kim M. Knyal for significant contributions to the discussions of pre- and post-load processing and project managing the ETL process, respectively.

We had an extraordinary team of reviewers who crawled over the first version of the manuscript and made many helpful suggestions. It is always daunting to make significant changes to a manuscript that is "done" but this kind of deep review has been a tradition with the Toolkit series of books and was successful again this time. In alphabetic order, the reviewers included: Wouleta Ayele, Bob Becker, Jan-Willem Beldman, Ivan Chong, Maurice Frank, Mark Hodson, Paul Hoffman, Qi Jin, David Lyle, Michael Martin, Joy Mundy, Rostislav Portnoy, Malathi Vellanki, Padmini Ramanujan, Margy Ross, Jack Serra-Lima, and Warren Thornthwaite.

We owe special thanks to our spouses Robin Caserta and Julie Kimball for their support throughout this project and our children Tori Caserta, Brian Kimball, Sara (Kimball) Smith, and grandchild(!) Abigail Smith who were very patient with the authors who always seemed to be working.

Finally, the team at Wiley Computer books has once again been a real asset in getting this book finished. Thank you Bob Elliott, Kevin Kent, and Adaobi Obi Tulton.

About the Authors

Ralph Kimball, Ph.D., founder of the Kimball Group, has been a leading visionary in the data warehouse industry since 1982 and is one of today's most well-known speakers, consultants, teachers, and writers. His books include *The Data Warehouse Toolkit* (Wiley, 1996), *The Data Warehouse Lifecycle Toolkit* (Wiley, 1998), *The Data Webhouse Toolkit* (Wiley, 2000), and *The Data Warehouse Toolkit, Second Edition* (Wiley, 2002). He also has written for *Intelligent Enterprise* magazine since 1995, receiving the Readers' Choice Award since 1999.

Ralph earned his doctorate in electrical engineering at Stanford University with a specialty in man-machine systems design. He was a research scientist, systems development manager, and product marketing manager at Xerox PARC and Xerox Systems' Development Division from 1972 to 1982. For his work on the Xerox Star Workstation, the first commercial product with windows, icons, and a mouse, he received the Alexander C. Williams award from the IEEE Human Factors Society for systems design. From 1982 to 1986 Ralph was Vice President of Applications at Metaphor Computer Systems, the first data warehouse company. At Metaphor, Ralph invented the "capsule" facility, which was the first commercial implementation of the graphical data flow interface now in widespread use in all ETL tools. From 1986 to 1992 Ralph was founder and CEO of Red Brick Systems, a provider of ultra-fast relational database technology dedicated to decision support. In 1992 Ralph founded Ralph Kimball Associates, which became known as the Kimball Group in 2004. The Kimball Group is a team of highly experienced data warehouse design professionals known for their excellence in consulting, teaching, speaking, and writing.

Joe Caserta is the founder and Principal of Caserta Concepts, LLC. He is an influential data warehousing veteran whose expertise is shaped by years of industry experience and practical application of major data warehousing tools and databases. Joe is educated in Database Application Development and Design, Columbia University, New York.

Introduction

The Extract-Transform-Load (ETL) system is the foundation of the data warehouse. A properly designed ETL system extracts data from the source systems, enforces data quality and consistency standards, conforms data so that separate sources can be used together, and finally delivers data in a presentation-ready format so that application developers can build applications and end users can make decisions. This book is organized around these four steps.

The ETL system makes or breaks the data warehouse. Although building the ETL system is a *back room* activity that is not very visible to end users, it easily consumes 70 percent of the resources needed for implementation and maintenance of a typical data warehouse.

The ETL system adds significant value to data. It is far more than plumbing for getting data out of source systems and into the data warehouse. Specifically, the ETL system:

- Removes mistakes and corrects missing data
- Provides documented measures of confidence in data
- Captures the flow of transactional data for safekeeping
- Adjusts data from multiple sources to be used together
- Structures data to be usable by end-user tools

ETL is both a simple and a complicated subject. Almost everyone understands the basic mission of the ETL system: to get data out of the source and load it into the data warehouse. And most observers are increasingly appreciating the need to clean and transform data along the way. So much for the simple view. It is a fact of life that the next step in the design of

the ETL system breaks into a thousand little subcases, depending on your own weird data sources, business rules, existing software, and unusual destination-reporting applications. The challenge for all of us is to tolerate the thousand little subcases but to keep perspective on the simple overall mission of the ETL system. Please judge this book by how well we meet this challenge!

The Data Warehouse ETL Toolkit is a practical guide for building successful ETL systems. This book is not a survey of all possible approaches! Rather, we build on a set of consistent techniques for delivery of dimensional data. Dimensional modeling has proven to be the most predictable and cost effective approach to building data warehouses. At the same time, because the dimensional structures are the same across many data warehouses, we can count on reusing code modules and specific development logic.

This book is a roadmap for planning, designing, building, and running the back room of a data warehouse. We expand the traditional ETL steps of extract, transform, and load into the more actionable steps of extract, clean, conform, and deliver, although we resist the temptation to change ETL into ECCD!

In this book, you'll learn to:

- Plan and design your ETL system
- Choose the appropriate architecture from the many possible choices
- Manage the implementation
- Manage the day-to-day operations
- Build the development/test/production suite of ETL processes
- Understand the tradeoffs of various back-room data structures, including flat files, normalized schemas, XML schemas, and star join (dimensional) schemas
- Analyze and extract source data
- Build a comprehensive data-cleaning subsystem
- Structure data into dimensional schemas for the most effective delivery to end users, business-intelligence tools, data-mining tools, OLAP cubes, and analytic applications
- Deliver data effectively both to highly centralized and profoundly distributed data warehouses using the same techniques
- Tune the overall ETL process for optimum performance

The preceding points are many of the big issues in an ETL system. But as much as we can, we provide lower-level technical detail for:

- Implementing the key enforcement steps of a data-cleaning system for column properties, structures, valid values, and complex business rules
- Conforming heterogeneous data from multiple sources into standardized dimension tables and fact tables
- Building replicatable ETL modules for handling the natural time variance in dimensions, for example, the three types of slowly changing dimensions (SCDs)
- Building replicatable ETL modules for multivalued dimensions and hierarchical dimensions, which both require associative bridge tables
- Processing extremely large-volume fact data loads
- Optimizing ETL processes to fit into highly constrained load windows
- Converting batch and file-oriented ETL systems into continuously streaming real-time ETL systems

For illustrative purposes, Oracle is chosen as a common dominator when specific SQL code is revealed. However, similar code that presents the same results can typically be written for DB2, Microsoft SQL Server, or any popular relational database system.

And perhaps as a side effect of all of these specific recommendations, we hope to share our enthusiasm for developing, deploying, and managing data warehouse ETL systems.

Overview of the Book: Two Simultaneous Threads

Building an ETL system is unusually challenging because it is so heavily constrained by unavoidable realities. The ETL team must live with the business requirements, the formats and deficiencies of the source data, the existing legacy systems, the skill sets of available staff, and the ever-changing (and legitimate) needs of end users. If these factors aren't enough, the budget is limited, the processing-time windows are too narrow, and important parts of the business come grinding to a halt if the ETL system doesn't deliver data to the data warehouse!

Two simultaneous threads must be kept in mind when building an ETL system: the Planning & Design thread and the Data Flow thread. At the highest level, they are pretty simple. Both of them progress in an orderly fashion from left to right in the diagrams. Their interaction makes life very

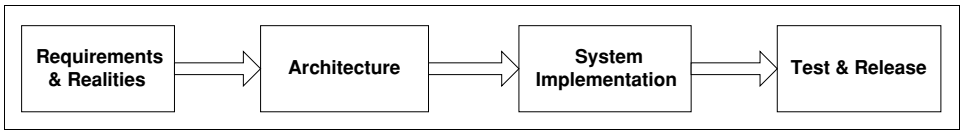

Figure Intro-1 The Planning and Design Thread.

interesting. In Figure Intro-1 we show the four steps of the Planning & Design thread, and in Figure Intro-2 we show the four steps of the Data Flow thread.

To help you visualize where we are in these two threads, in each chapter we call out process checks. The following example would be used when we are discussing the requirements for data cleaning:

PROCESS CHECK **Planning & Design:**
Requirements/Realities **→ Architecture → Implementation → Test/Release**

Data Flow: Extract → *Clean* → Conform → Deliver

The Planning & Design Thread

The first step in the Planning & Design thread is accounting for all the *requirements and realities*. These include:

- Business needs
- Data profiling and other data-source realities
- Compliance requirements
- Security requirements
- Data integration
- Data latency
- Archiving and lineage

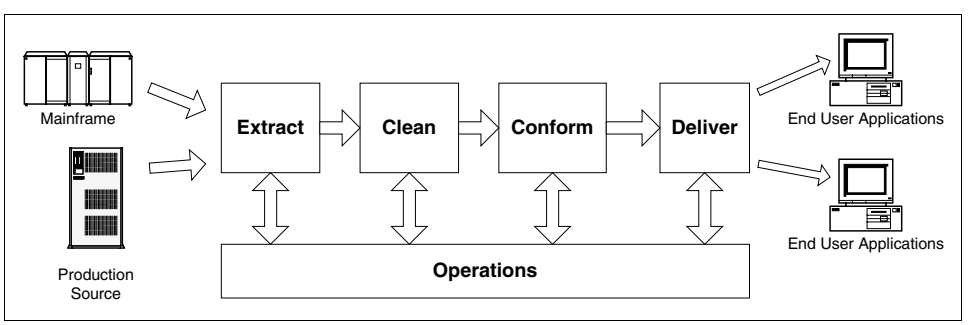

Figure Intro-2 The Data Flow Thread.

- End user delivery interfaces
- Available development skills
- Available management skills
- Legacy licenses

We expand these individually in the Chapter 1, but we have to point out at this early stage how much each of these bullets affects the nature of your ETL system. For this step, as well as all the steps in both major threads, we point out the places in this book when we are talking specifically about the given step.

The second step in this thread is the *architecture* step. Here is where we must make big decisions about the way we are going to build our ETL system. These decisions include:

- Hand-coded versus ETL vendor tool
- Batch versus streaming data flow
- Horizontal versus vertical task dependency
- Scheduler automation
- Exception handling
- Quality handling
- Recovery and restart
- Metadata
- Security

The third step in the Planning & Design thread is *system implementation*. Let's hope you have spent some quality time on the previous two steps before charging into the implementation! This step includes:

- Hardware
- Software
- Coding practices
- Documentation practices
- Specific quality checks

The final step sounds like administration, but the design of the test and release procedures is as important as the more tangible designs of the preceding two steps. Test and release includes the design of the:

- Development systems
- Test systems

- Production systems
- Handoff procedures
- Update propagation approach
- System snapshoting and rollback procedures
- Performance tuning

The Data Flow Thread

The Data Flow thread is probably more recognizable to most readers because it is a simple generalization of the old E-T-L extract-transform-load scenario. As you scan these lists, begin to imagine how the Planning & Design thread affects each of the following bullets. The *extract* step includes:

- Reading source-data models
- Connecting to and accessing data
- Scheduling the source system, intercepting notifications and daemons
- Capturing changed data
- Staging the extracted data to disk

The *clean* step involves:

- Enforcing column properties
- Enforcing structure
- Enforcing data and value rules
- Enforcing complex business rules
- Building a metadata foundation to describe data quality
- Staging the cleaned data to disk

This step is followed closely by the *conform* step, which includes:

- Conforming business labels (in dimensions)
- Conforming business metrics and performance indicators (in fact tables)
- Deduplicating
- Householding
- Internationalizing
- Staging the conformed data to disk

Finally, we arrive at the payoff step where we *deliver* our wonderful data to the end-user application. We spend most of Chapters 5 and 6 on delivery techniques because, as we describe in Chapter 1, you still have to serve the food after you cook it! Data delivery from the ETL system includes:

- Loading flat and snowflaked dimensions
- Generating time dimensions
- Loading degenerate dimensions
- Loading subdimensions
- Loading types 1, 2, and 3 slowly changing dimensions
- Conforming dimensions and conforming facts
- Handling late-arriving dimensions and late-arriving facts
- Loading multi-valued dimensions
- Loading ragged hierarchy dimensions
- Loading text facts in dimensions
- Running the surrogate key pipeline for fact tables
- Loading three fundamental fact table grains
- Loading and updating aggregations
- Staging the delivered data to disk

In studying this last list, you may say, "But most of that list is modeling, not ETL. These issues belong in the front room." We respectfully disagree. In our interviews with more than 20 data warehouse teams, more than half said that the design of the ETL system took place at the same time as the design of the target tables. These folks agreed that there were two distinct roles: data warehouse architect and ETL system designer. But these two roles often were filled by the same person! So this explains why this book carries the data all the way from the original sources into each of the dimensional database configurations.

The basic four-step data flow is overseen by the *operations* step, which extends from the beginning of the extract step to the end of the delivery step. Operations includes:

- Scheduling
- Job execution
- Exception handling
- Recovery and restart
- Quality checking

- Release
- Support

Understanding how to think about these two fundamental threads (Planning & Design and Data Flow) is the real goal of this book.

How the Book Is Organized

To develop the two threads, we have divided the book into four parts:

I. Requirements, Realities and Architecture
II. Data Flow
III. Implementation and Operations
IV. Real Time Streaming ETL Systems

This book starts with the requirements, realities, and architecture steps of the planning & design thread because we must establish a logical foundation for the design of any kind of ETL system. The middle part of the book then traces the entire data flow thread from the extract step through to the deliver step. Then in the third part we return to implementation and operations issues. In the last part, we open the curtain on the exciting new area of real time streaming ETL systems.

Part I: Requirements, Realities, and Architecture

Part I sets the stage for the rest of the book. Even though most of us are eager to get started on moving data into the data warehouse, we have to step back to get some perspective.

Chapter 1: Surrounding the Requirements

The ETL portion of the data warehouse is a classically overconstrained design challenge. In this chapter we put some substance on the list of requirements that we want you to consider up front before you commit to an approach. We also introduce the main architectural decisions you must take a stand on (whether you realize it or not).

This chapter is the right place to define, as precisely as we can, the major vocabulary of data warehousing, at least as far as this book is concerned. These terms include:

- Data warehouse
- Data mart

- ODS (operational data store)
- EDW (enterprise data warehouse)
- Staging area
- Presentation area

We describe the mission of the data warehouse as well as the mission of the ETL team responsible for building the *back room* foundation of the data warehouse. We briefly introduce the basic four stages of Data Flow: extracting, cleaning, conforming, and delivering. And finally we state as clearly as possible why we think dimensional data models are the keys to success for every data warehouse.

Chapter 2: ETL Data Structures

Every ETL system must stage data in various permanent and semipermanent forms. When we say *staging*, we mean writing data to the disk, and for this reason the ETL system is sometimes referred to as the staging area. You might have noticed that we recommend at least some form of staging after each of the major ETL steps (extract, clean, conform, and deliver). We discuss the reasons for various forms of staging in this chapter.

We then provide a systematic description of the important data structures needed in typical ETL systems: flat files, XML data sets, independent DBMS working tables, normalized entity/relationship (E/R) schemas, and dimensional data models. For completeness, we mention some special tables including legally significant audit tracking tables used to prove the provenance of important data sets, as well as mapping tables used to keep track of surrogate keys. We conclude with a survey of metadata typically surrounding these types of tables, as well as naming standards. The metadata section in this chapter is just an introduction, as metadata is an important topic that we return to many times in this book.

Part II: Data Flow

The second part of the book presents the actual steps required to effectively extract, clean, conform, and deliver data from various source systems into an ideal dimensional data warehouse. We start with instructions on selecting the system-of-record and recommend strategies for analyzing source systems. This part includes a major chapter on building the cleaning and conforming stages of the ETL system. The last two chapters then take the cleaned and conformed data and repurpose it into the required dimensional structures for delivery to the end-user environments.

Chapter 3: Extracting

This chapter begins by explaining what is required to design a logical data mapping after data analysis is complete. We urge you to create a logical data map and to show how it should be laid out to prevent ambiguity in the mission-critical specification. The logical data map provides ETL developers with the functional specifications they need to build the physical ETL process.

A major responsibility of the data warehouse is to provide data from various legacy applications throughout the enterprise data in a single cohesive repository. This chapter offers specific technical guidance for integrating the heterogeneous data sources found throughout the enterprise, including mainframes, relational databases, XML sources, flat files, Web logs, and enterprise resource planning (ERP) systems. We discuss the obstacles encountered when integrating these data sources and offer suggestions on how to overcome them. We introduce the notion of conforming data across multiple potentially incompatible data sources, a topic developed fully in the next chapter.

Chapter 4: Cleaning and Conforming

After data has been extracted, we subject it to cleaning and conforming. *Cleaning* means identifying and fixing the errors and omissions in the data. *Conforming* means resolving the labeling conflicts between potentially incompatible data sources so that they can be used together in an enterprise data warehouse.

This chapter makes an unusually serious attempt to propose specific techniques and measurements that you should implement as you build the cleaning and conforming stages of your ETL system. The chapter focuses on data-cleaning objectives, techniques, metadata, and measurements.

In particular, the techniques section surveys the key approaches to data profiling and data cleaning, and the measurements section gives examples of how to implement data-quality checks that trigger alerts, as well as how to provide guidance to the data-quality steward regarding the overall health of the data.

Chapter 5: Delivering Dimension Tables

This chapter and Chapter 6 are the payoff chapters in this book. We believe that the whole point of the data warehouse is to deliver data in a simple, actionable format for the benefit of end users and their analytic applications. Dimension tables are the context of a business' measurements. They are also the entry points to the data because they are the targets for almost all data