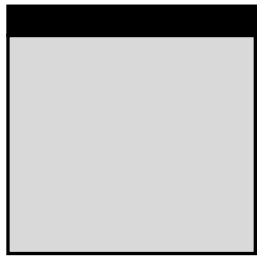


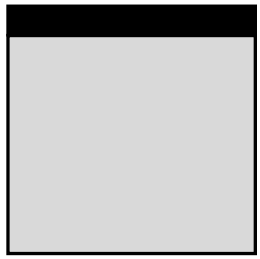
Mastering Data Warehouse Aggregates

Solutions for Star Schema Performance

Christopher Adamson

Wiley Publishing, Inc.

Mastering Data Warehouse Aggregates



Mastering Data Warehouse Aggregates

Solutions for Star Schema Performance

Christopher Adamson

Wiley Publishing, Inc.

Mastering Data Warehouse Aggregates: Solutions for Star Schema Performance
Published by
Wiley Publishing, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Published simultaneously in Canada

ISBN-13: 978-0-471-77709-0
ISBN-10: 0-471-77709-9

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

1MA/SQ/QW/QW/IN

Library of Congress Cataloging-in-Publication Data

Adamson, Christopher, 1967–
Mastering data warehouse aggregates: solutions for star schema performance / Christopher Adamson.
p. cm.
Includes index.
ISBN-13: 978-0-471-77709-0 (pbk.)
ISBN-10: 0-471-77709-9 (pbk.)
1. Data warehousing. I. Title.
QA76.9.D37A333 2006
005.74—dc22

2006011219

For Wayne H. Adamson

1929–2003

Through those whose lives you touched,
your spirit of love endures.

About the Author

Christopher Adamson is a data warehousing consultant and founder of Oakton Software LLC. An expert in star schema design, he has managed and executed data warehouse implementations in a variety of industries. His customers have included Fortune 500 companies, large and small businesses, government agencies, and data warehousing tool vendors. Mr. Adamson also teaches dimensional modeling and is a co-author of *Data Warehouse Design Solutions* (also from Wiley). He can be contacted through his website, `www.ChrisAdamson.net`.

Credits

Executive Editor
Robert Elliott

Development Editor
Brian Herrmann

Technical Editor
Jim Hadley

Copy Editor
Nancy Rapoport

Editorial Manager
Mary Beth Wakefield

Production Manager
Tim Tate

Vice President and Executive Group Publisher
Richard Swadley

Vice President and Executive Publisher
Joseph B. Wikert

Project Coordinator
Michael Kruzil

Graphics and Production Specialists
Jennifer Click
Denny Hager
Stephanie D. Jumper
Heather Ryan

Quality Control Technicians
John Greenough
Brian H. Walls

Proofreading and Indexing
Techbooks

Contents

Foreword

In 1998 I wrote the foreword for Chris Adamson and Mike Venerable's book *Data Warehouse Design Solutions* (Wiley, 1998). Over the intervening eight years I have been delighted to track that book, as it has stayed high in the list of data warehouse best sellers, even through today. Chris and Mike had identified a set of data warehouse design challenges and were able to speak very effectively in that book to the community of data warehouse designers.

Viewed in the right perspective, the mission of data warehousing has not changed at all since 1998! In that foreword, I wrote that the data warehouse must be driven from business analysis needs, must be a mirror of management's urgent priorities, and must be a presentation facility that is understandable and fast. All of these perspectives have held true through today. While our databases have exploded in size, and the database content has become much more operational, the original description of the data warehouse rings true. If anything, the data warehouse, in its role as the platform for all forms of business intelligence, has become much more important than it was in 1998.

At the same time that the reach of the data warehouse has penetrated to every worker's desktop, we have all been swept along by the development of the Internet, and particularly search engines like Google. This parallel revolution, surprisingly, has sent data warehousing and business intelligence a powerful and simple message. As the saying goes, "The medium is the message." In this case, Google's message is:

You can search the entire contents of the Internet in less than a second.

The message to data warehousing is:

You should expect instantaneous results from your data warehouse queries.

To be perfectly frank, data warehousing and business intelligence have so far made only partial progress toward instantaneous performance. Our databases are more complicated than Google's documents, and our queries are more complex. *But*, we have some powerful tools that can be used to get us much closer to the goal of instantaneous performance.

Those of us who, like Chris and the Kimball Group, have long recognized that the class of data warehouse designs known as dimensional models offers a systematic opportunity for a huge performance boost, above and beyond database indexes, hardware RAM, faster processors, or parallelism. In fact, this additional performance opportunity, known as *aggregates*, when used correctly, can trump all the other performance techniques!

The idea behind aggregates is very simple. Always start with the most atomic, transaction-grained data available from the original source systems. Place that atomic data in full view of the end users in a dimensional format. Of course, if you stop there, you will have performance problems because many queries will do a huge amount of I/O no matter how much hardware you throw at the problem. Now aggregates come to the rescue. You systematically create a set of physically stored, pre-calculated summary records that are predictable common queries, or parts of queries posed by the end users. These summary records are the aggregates.

Aggregates, when used correctly, can provide performance improvements of a hundred or even a thousand times. No other technology is capable of such gains.

This book is all about aggregates. Chris explains how they rely on the dimensional approach, which aggregates to build, how to build them, and how to maintain them. He also shows in detail how Oracle's materialized views and IBM's materialized query tables are perfect examples of aggregates used effectively.

I was delighted to see Chris return to being an author after his wonderful first book. His only excuse for waiting eight years was that he was "busy building data warehouses." I'll accept that excuse! Now we can apply Chris's insights into making our data warehouse and business intelligence systems a big step closer to being instantaneous.

Ralph Kimball
Founder, Kimball Group
Boulder Creek, California

Acknowledgments

Thank you to everyone who read my first book, *Data Warehouse Design Solutions*, which I wrote with Mike Venerable. The positive feedback we received from around the world was unexpected, and most appreciated. Without your warm reception, I doubt that the current volume would have come to pass.

This book would not have been possible without Ralph Kimball. The value of his contribution to the data warehousing world cannot be understated. He has established a practical and powerful approach to data warehousing and provided terminology and principles for dimensional modeling that are used throughout the industry. I am deeply grateful for Ralph's continued support and encouragement, without which neither this nor my previous book would have been written.

I thank everyone at Wiley who contributed to this effort. Bob Elliott was a pleasure to work with and provided constructive criticism that was instrumental in shaping this book. Brian Herrmann made the writing process as painless as possible. I also thank the anonymous reviewers of my original outline, whose comments made this a better book.

Thanks also to Jim Hadley, who put in long hours reviewing drafts of this book. Through his detailed comments and advice, he made a substantial contribution to this effort. His continuing encouragement got me through several rough spots.

I am grateful to the customers and colleagues with whom I have worked over the years. The opportunity to learn from one another enriches us all. In particular, I thank three people as yet unmentioned. Mike Venerable has offered me opportunities that have shaped my career, along with guidance and advice that have helped me grow in numerous dimensions. Greg Jones's

work managing data warehouse projects has profoundly influenced my own perspective, as is evident in Chapter 7. And Randall Porter has always been a welcome source of professional guidance, which was offered over many breakfasts during the writing of this book.

A very special thank you to my wife, Gladys, and sons, Justin and Carter, whose support and encouragement gave me the resolve I needed to complete this project. I also received support from my mother, sister, in-laws, and sisters-in-law. I could not have done this without all of you.

Introduction

In the battle to improve data warehouse performance, no weapon is more powerful and efficient than the aggregate table. A well-planned set of aggregates can have an extraordinary impact on the overall throughput of the data warehouse. After you ensure that the database is properly designed, configured, and tuned, any measures taken to address data warehouse performance should begin with aggregates.

Yet many businesses continue to ignore aggregates, turning instead to proprietary hardware products, converting to specialized databases, or implementing complex caching architectures. These solutions carry high price tags for acquisition and implementation and often require specialized skills to maintain. This book aims to fill the knowledge gap that has led businesses down this expensive and risky path.

In these pages, you will find tools and techniques you can use to bring stunning performance gains to *your* data warehouse. This book develops a set of best practices for the selection, design, construction, and use of aggregate tables. It explores how these techniques can be incorporated into projects, studies advanced design considerations, and covers how aggregates affect other aspects of the data warehouse lifecycle.

Intended Audience

This book is intended for *you*, the data warehouse practitioner with an interest in improving query performance. You may serve any one of a number of roles in the data warehouse environment, including:

- Business analyst
- Star schema designer
- Report developer
- ETL developer
- Project manager
- Database administrator
- Data administrator
- I.T. director
- Chief information officer
- Power user

Regardless of your role or current level of knowledge, the best practices in this book will help you and your team achieve astounding increases in data warehouse performance, without requiring an investment in exotic technologies.

It will be assumed that you have a very basic familiarity with relational database technology, understanding the concepts of tables, columns, and joins. Occasional examples of SQL code will be provided, and they will be fully explained in the accompanying text.

For those new to data warehousing, the background necessary to understand the examples will be provided along the way. For example, an overview of the star schema is presented in Chapter 1. The Extract Transform Load (ETL) process for the data warehouse is described in Chapter 5. The high-level data mart implementation process is described in Chapter 7.

About This Book

This book assumes a star schema approach to data warehousing. The necessary background is provided for readers new to this approach. It also considers implications of snowflake designs and, to a lesser extent, schemas in third normal form (3NF).

The design principles and best practices developed in each chapter make no assumptions about specific software products in the data warehouse. This tool-agnostic perspective is periodically supplemented with specific advice for users of Oracle's materialized views and IBM DB/2's materialized query tables.

Star Schema Approach

The techniques presented in this book are intended for data warehouses that are designed according to the principles of *dimensional modeling*, more popularly known as the *star schema approach*. Popularized by Ralph Kimball in the 1990s, the dimensional model is now widely accepted as the optimal method to organize information for analytic consumption.

Ralph Kimball and Margy Ross provide a comprehensive treatment of dimensional modeling in *The Data Warehouse Toolkit, Second Edition* (Wiley, 2002). The seminal work on the subject, their book is required reading for any student of data warehousing. The best practices in this book build on the foundation provided by Kimball and Ross and are described using terminology established by *The Toolkit*.

If you are not familiar with the star schema approach to data warehouse design, Chapter 1 provides an overview of the basic principles necessary to understand the examples in this book.

Snowflakes and 3NF Designs

Although this book focuses on the star schema, it does not ignore other approaches to schema design. From time to time, this book will examine the impact of a *snowflake design* on principles established throughout the book. For example, implications of a snowflake schema for aggregate design are explored in Chapters 2 and 3, and discussed more fully in Chapter 8.

In addition, Chapter 8 will look at how dimensional aggregates can service a *third normal form* schema design. Because of the complex relationships between the tables of a normalized schema, dimensional aggregates can have a tremendous impact. Of course, this is really the impact of the dimensional model itself. Best practices would suggest beginning with the most granular design possible, which is not really an aggregate at all. Still, a dimensional perspective can be used to augment query performance in such an environment.

Tool Independence

This book makes no assumptions regarding the presence of specific software products in your data warehouse architecture. Many commercial products offer features to assist in the implementation of aggregate tables. Each implementation is different; each has its own benefits and drawbacks; all are constantly changing.

Regardless of the tools used to build and navigate aggregates, you will need to address the same major tasks. You must choose which aggregates to implement; the aggregates must be designed; the aggregates must be built; a process must be established to ensure they are refreshed, or loaded, on a regular basis; the warehouse must be configured so that application queries are redirected to the aggregates.

This book provides a set of principles and best practices to guide you through these common tasks.

You can also use the principles in this book to guide the selection of specific technologies. For example, one component that you may need to add to your data warehouse architecture is the *aggregate navigator*. Chapter 4 develops a set of requirements for the aggregate navigator function. Three styles of commercial implementations are identified and evaluated against these requirements. You can use these requirements to evaluate your current technology options, as described in Chapter 7. They will remain valid even as specific products change and evolve.

Materialized Views and Materialized Query Tables

Specific database features from Oracle (materialized views) and IBM's DB/2 (materialized query tables) can be used to load and maintain aggregate tables as well as provide aggregate navigation services.

Throughout this book, the impact of using these technologies to build and navigate dimensional aggregates is explored. After establishing principles and best practices, we consider the implications of using these products. What is potentially gained or lost? How can you modify your process to accommodate the products' strengths and weaknesses? This is information that cannot be gleaned from a syntax reference manual.

Keep in mind that these products continue to evolve. Over time, their capabilities can be expected to expand and change. If you use these products, it behooves you to study their capabilities closely, compare them with the requirements of dimensional aggregation, test their application, and identify relevant implications. In fact, this is advised for users of *any* tool in Chapter 7.

MATERIALIZED VIEWS AND MATERIALIZED QUERY TABLES

The tool-agnostic principles and techniques in this book are periodically supplemented with a look at the impact of Oracle's *materialized views* and IBM DB/2's *materialized query tables*.

TOPIC	CHAPTER	DESCRIPTION
Aggregate Design	Chapter 3	Schema designers targeting these technologies should model the hierarchies implicit within a dimension table, and the relationships among their attributes. This information should be included in design documentation, along with defining queries for each aggregate table.
Aggregate Use	Chapter 4	Chapter 4 describes the use of materialized views and materialized query tables to provide query rewrite capabilities. It also shows how these technologies are used to implement aggregate fact tables, virtual aggregate dimension tables, and pre-joined aggregates.
Aggregate Refresh	Chapter 5	Materialized views and materialized query tables do not eliminate the need to manage the refresh of aggregates. It is also necessary to coordinate the refresh mechanism with the query rewrite mechanism.
Aggregate Construction	Chapter 6	It is not necessary to build an ETL process to load a materialized view or materialized query table. Once their refresh is configured, the database will take care of this job. But some adjustments to the base schema's ETL process may improve the overall performance of the aggregates.

Purpose of Each Chapter

This book is organized into chapters that address the major activities involved in the implementation of star schema aggregates. After establishing some fundamentals, chapters are dedicated to aggregate selection, design, usage, and construction. The remaining chapters address the organization of these activities into project plans, explore advanced design considerations, and address other impacts on the data warehouse.

Chapter 1: Fundamentals of Aggregates

This chapter establishes a foundation on which the rest of the book will build. It introduces the *star schema*, *aggregate tables*, and the *aggregate navigator*. Even if you are already familiar with these concepts, you should read Chapter 1. It establishes guiding principles for the development of *invisible aggregates*, which have zero impact on production applications. These principles will shape the best practices developed through the rest of the book. This chapter also introduces several forms of summarization that are not invisible to applications but may provide useful performance benefits.

Chapter 2: Choosing Aggregates

Chapter 2 takes on the difficult process of determining which aggregates should be built. You will learn how to identify and describe potential aggregates and determine the appropriate combination for implementation. This will require balancing the performance of potential aggregates with their potential usage and available resources. A variety of techniques will prove useful in identifying high-value aggregate tables.

Chapter 3: Designing Aggregates

The design of aggregate tables requires the same rigor as that of the base schema. Chapter 3 lays out a detailed set of principles for the design of dimensional aggregates. Best practices are identified and explained in detail, and a concrete set of deliverables is developed for the design process. Common pitfalls that can disrupt accuracy or ease of use are fully explored.

Chapter 4: Using Aggregates

In the most successful implementations, aggregate tables are invisible to users and applications. The job of the aggregate navigator is to redirect all queries to the best performing summaries. Chapter 4 develops a set of requirements for the aggregate navigator and uses them to evaluate three common styles of solutions. It explores two specific technologies in detail—Oracle's materialized views and IBM DB/2's materialized query tables—and provides practical advice for working without an aggregate navigator.

Chapter 5: ETL Part 1: Incorporating Aggregates

This book dedicates two chapters to the process of building aggregate tables. Chapter 5 describes how the base schema is loaded and how aggregates are integrated into that process. You will learn when it makes sense to design an incremental load for aggregate tables, and when you are better off dropping and rebuilding them each time the base schema is updated. For data warehouses loaded during batch windows, this chapter outlines several benefits of loading aggregates after the base schema. The ETL process will be required to interact with the aggregate navigator, or to take the entire data warehouse offline during the load. Data warehouses loaded in real-time require a different strategy for the maintenance of aggregates; specific techniques are discussed to minimize the impact of aggregates on this process.

Database features such as materialized views or materialized query tables may automate the construction process but are subject to the same requirements. As Chapter 5 shows, they must be configured to remain synchronized with the base schema, and designers must still choose between drop-and-rebuild and incremental load.

Chapter 6: ETL Part 2: Loading Aggregates

The second of two chapters on ETL, Chapter 6 describes the specific tasks required to load aggregate tables. Best practices are provided for identifying changed data in the base schema, constructing aggregate dimensions and their surrogate keys, and building aggregate fact tables. Pre-joined aggregates are also considered, along with complications that can arise from the presence of type 1 attributes.

The best practices in this chapter apply whether the load is developed using an ETL tool, or hand-coded. Database features such as materialized views or materialized query tables eliminate the need to design load routines, but may benefit from some adjustment to the schema design.

Chapter 7: Aggregates and Your Project

Aggregates should always be designed and implemented as part of a project. Chapter 7 provides a standard set of tasks and deliverables that can be used to add aggregates to existing schema, or to incorporate aggregates into the scope of a larger data warehouse development project. Major project phases are covered, including strategy, design, construction, testing, and deployment. The ongoing maintenance of aggregates is discussed, tying specific responsibilities to established data warehousing roles.

Chapter 8: Advanced Aggregate Design

This chapter outlines numerous advanced techniques for star schema design and fully analyzes the implications of each technique on aggregation. Design topics include:

- The periodic snapshot
- The accumulating snapshot
- Two kinds of factless fact tables
- Three kinds of bridge tables
- The transaction dimension
- Families of core and custom schemas

Chapter 8 also looks at how the techniques in this book can be adapted for snowflake schemas and third normal form designs.

Chapter 9: Related Topics

This final chapter collects several remaining topics that are influenced by aggregates:

- *The archive process* must be extended to involve aggregate tables. Some common misconceptions are discussed, and often-overlooked opportunities are highlighted.
- *Security requirements* may call for special care in implementing aggregates, which may also prove part of the solution.
- *Derived tables* are summarizations of base schema data that are not invisible. They include merged fact tables, sliced fact tables, and pivoted fact tables. Standard invisible aggregates may further summarize derived tables.
- *Deploying summary data before detail* can present new challenges, particularly if unanticipated. This chapter concludes by providing alternative techniques to deal with this unusual problem.

Glossary

Important terms used throughout this book are collected and defined in the glossary. You may find it useful to refer to these definitions as you read this book, particularly if you choose to read the chapters out of sequence.

Mastering Data Warehouse Aggregates

CHAPTER 1

Fundamentals of Aggregates

A decade ago, Ralph Kimball described aggregate tables as "the single most dramatic way to improve performance in a large data warehouse." Writing in *DBMS Magazine* ("Aggregate Navigation with (Almost) No Metadata," August 1996), Kimball continued:

> *Aggregates can have a very significant effect on performance, in some cases speeding queries by a factor of one hundred or even one thousand. No other means exist to harvest such spectacular gains.*

This statement rings as true today as it did ten years ago. Since then, advances in hardware and software have dramatically improved the capacity and performance of the data warehouse. Aggregates *compound* the effect of these improvements, providing performance gains that fully harness capabilities of the underlying technologies.

And the pressure to improve data warehouse performance is as strong as ever. As the baseline performance of underlying technologies has improved, warehouse developers have responded by storing and analyzing larger and more granular volumes of data. At the same time, warehouse systems have been opened to larger numbers of users, internal and external, who have come to expect instantaneous access to information.

This book empowers *you* to address these pressures. Using aggregate tables, you can achieve an extraordinary improvement in the speed of *your* data warehouse. And you can do it today, without making expensive upgrades to hardware, converting to a new database platform, or investing in exotic and proprietary technologies.

Although aggregates can have a powerful impact on data warehouse performance, they can also be misused. If not managed carefully, they can cause confusion, impose inordinate maintenance requirements, consume massive amounts of storage, and even provide inaccurate results. By following the best practices developed in this book, you can avoid these outcomes and maximize the positive impact of aggregates.

The introduction of aggregate tables to the data warehouse will touch every aspect of the data warehouse lifecycle. A set of best practices governs their selection, design, construction, and usage. They will influence data warehouse planning, project scope, maintenance requirements, and even the archive process. Before exploring each of these topics, it is necessary to establish some fundamental principles and vocabulary.

This chapter establishes the foundation on which the rest of the book builds. It introduces the *star schema*, *aggregate tables*, and the *aggregate navigator*. Guiding principles are established for the development of *invisible aggregates*, which have zero impact on production applications—other than performance, of course. Last, this chapter explores several other forms of summarization that are not invisible to applications, but may also provide useful performance benefits.

Star Schema Basics

A star schema is a set of tables in a relational database that has been designed according to the principles of *dimensional modeling*. Ralph Kimball popularized this approach to data warehouse design in the 1990s. Through his work and writings, Kimball established standard terminology and best practices that are now used around the world to design and build data warehouse systems. With coauthor Margy Ross, he provides a detailed treatment of these principles in *The Data Warehouse Toolkit, Second Edition* (Wiley, 2002).

To follow the examples throughout this book, you must understand the fundamental principles of dimensional modeling. In particular, the reader must have a basic grasp of the following concepts:

- The differences between data warehouse systems and operational systems
- How facts and dimensions support the measurement of a business process
- The tables of a star schema (fact tables and dimension tables) and their purposes

- The purpose of surrogate keys in dimension tables
- The grain of a fact table
- The additivity of facts
- How a star schema is queried
- Drilling across multiple fact tables
- Conformed dimensions and the warehouse bus
- The basic architecture of a data warehouse, including ETL software and BI software

If you are familiar with these topics, you may wish to skip to the section "Invisible Aggregates," later in this chapter.

For everyone else, this section will bring you up-to-speed. Although not a substitute for Kimball and Ross's book, this overview provides the background needed to understand the examples throughout this book. I encourage *all* readers to read *The Toolkit* for more immersion in the principles of dimensional modeling, particularly anyone involved in the design of the dimensional data warehouse.

Data warehouse designers will also benefit from reading *Data Warehouse Design Solutions*, by Chris Adamson and Mike Venerable (Wiley, 1998). This book explores the application of these principles in the service of specific business objectives and covers standard business processes in a wide variety of industries.

Operational Systems and the Data Warehouse

Data warehouse systems and operational systems have fundamentally different purposes. An operational system supports the *execution* of business process, while the data warehouse supports the *evaluation* of the process. Their distinct purposes are reflected in contrasting usage profiles, which in turn suggest that different principles will guide their design. The principles of dimensional modeling are specifically adapted to the unique requirements of the warehouse system.

Operational Systems

An operational system directly supports the *execution* of business processes. By capturing detail about significant events or transactions, it constructs a record of the activity. A sales system, for example, captures information about orders, shipments, and returns; a human resources system captures information about the hiring and promotion of employees; an accounting system captures information about the management of the financial assets and liabilities of the business. Capturing the detail surrounding these activities is often so important that the operational system becomes a part of the process.