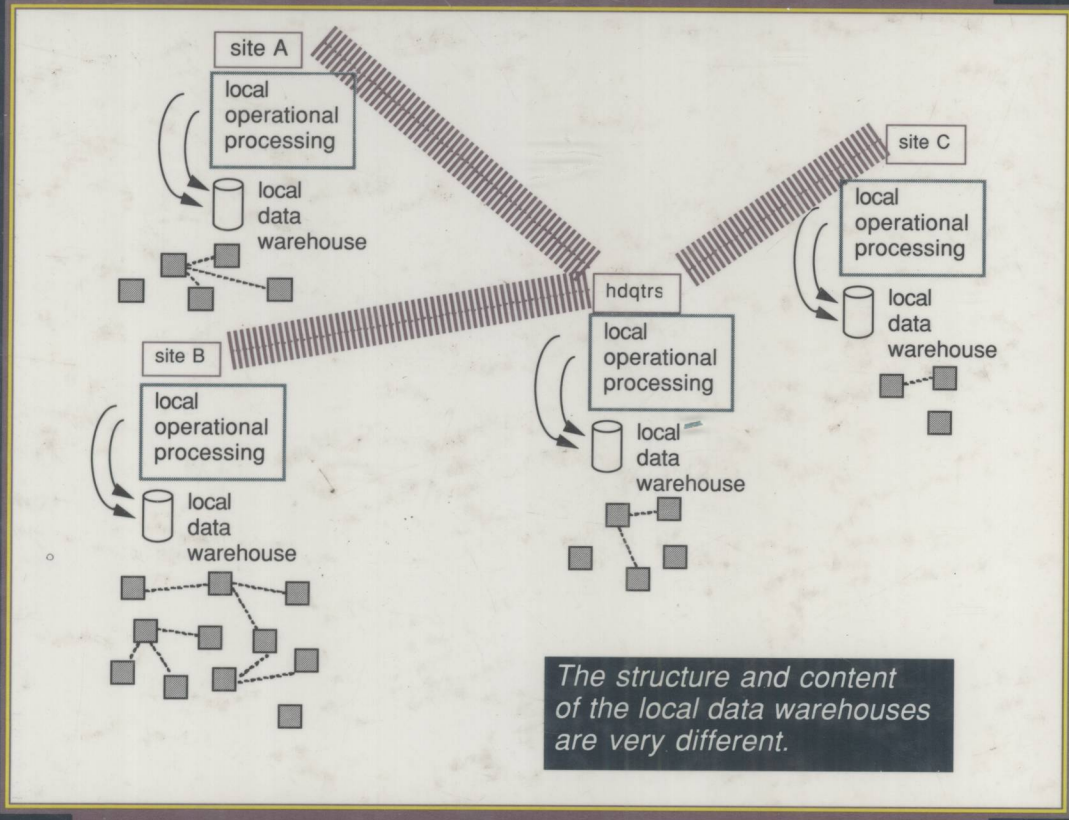
Building the Data Warehouse

W.H. Inmon



QED Publishing

Building the Data Warehouse

W.H. Inmon

QED Publishing Group
Boston • Toronto • London

P.O. Box 812070
Wellesley, MA 02181-0013

QED Publishing Group is a division of QED Information Sciences, Inc.

Library of Congress Catalog Number: 91-41284
International Standard Book Number: 0-89435-404-3

Printed in the United States of America

93 94 95 10 9 8 7 6 5 4 3 2 1

Library of Congress Cataloging-in-Publication Data
Inmon, William H.
Building the data warehouse / W.H. Inmon.
p. cm.
Includes bibliographical references and index.
ISBN 0-89435-404-3
1. Data base management. I. Title.
QA76.9.D3I5372 1992
005.74--dc20 91-41284
CIP

Building the Data Warehouse

Books from QED

Database

Managing IMS Databases
Building the Data Warehouse
Migrating to DB2
DB2: The Complete Guide to Implementation and Use
DB2 Design Review Guidelines
DB2: Maximizing Performance of Online Production Systems
Embedded SQL for DB2
SQL for DB2 and SQL/DS Application Developers
How to Use ORACLE SQL*PLUS
ORACLE: Building High Performance Online Systems
ORACLE Design Review Guidelines
Developing Client/Server Applications in an Architected Environment

Systems Engineering

From Mainframe to Workstations: Offloading Application Development
Software Configuration Management
On Time, Within Budget: Software Project Management Practices and Techniques
Information Systems Architecture: Development in the 90's
Quality Assurance for Information Systems
User-Interface Screen Design: Workstations, PC's, Mainframes
Managing Software Projects
The Complete Guide to Software Testing
A Structured Approach to Systems Testing
Rapid Application Prototyping
The Software Factory
Data Architecture: The Information Paradigm
Software Engineering with Formal Metrics
Using CASE Tools for Practical Management

Management

Enterprise Architecture Planning: Developing a Blueprint for Data, Applications, and Technology
Introduction to Data Security and Controls
How to Automate Your Computer Center
Controlling the Future
The UNIX Industry
Mind Your Business

IBM OS/2 Series

OS/2 Presentation Manager Programming for COBOL Programmers
Micro Focus Workbench for the Application Developer
OS/2 2.0: The Workplace Shell—A User's Guide and Tutorial

IBM Mainframe Series

VSE/SP and VSE/ESA: A Guide to Performance Tuning
CICS: A Guide to Application Debugging
CICS Application and System Programming
CICS: A Guide To Performance Tuning
MVS COBOL II Power Programmer's Desk Reference
VSE JCL and Subroutines for Application Programmers
VSE COBOL II Power Programmer's Desk Reference
Introduction to Cross System Product
Cross System Product Application Development
The MVS Primer
MVS/VSAM for the Application Programmer
TSO/E CLISTs: The Complete Tutorial and Desk Reference
CICS: A How-To for COBOL Programmers
QMF: How to Use Query Management Facility with DB2 and SQL/DS
DOS/VSE JCL: Mastering Job Control Language
DOS/VSE: CICS Systems Programming
VSAM: Guide to Optimization and Design
MVS/JCL: Mastering Job Control Language
MVS/TSO: Mastering CLISTs
MVS/TSO: Mastering Native Mode and ISPF
REXX in the TSO Environment, 2nd Edition

Technical

Rdb/VMS: Developing the Data Warehouse
AS/400 Architecture and Planning
C Language for Programmers
AS/400: A Practical Guide to Programming and Operations
Bean's Index to OSF/Motif, Xt Intrinsics, and Xlib Documentation for OSF/Motif Application Programmers
VAX/VMS: Mastering DCL Commands and Utilities
The PC Data Handbook
UNIX C Shell Desk Reference
Designing and Implementing Ethernet Networks
The Handbook for Microcomputer Technicians
Open Systems

QED books are available at special quantity discounts for educational uses, premiums, and sales promotions. Special books, book excerpts, and instructive materials can be created to meet specific needs.

This is Only a Partial Listing. For Additional Information or a Free Catalog contact

QED Information Sciences, Inc. • P. O. Box 812070 • Wellesley, MA 02181-0013

Telephone: 800-343-4848 or 617-237-5656 or fax 617-235-0826

For Jeanne Friedman,
a friend for all times

Preface

Databases and database theory have been around for a long time. Early renditions of databases centered around a single data base serving every purpose known to the information processing community—from transaction to batch processing to analytical processing. In most cases, the primary focus of the early database systems was operational—usually transactional—processing. In recent years, a more sophisticated notion of database has emerged—one that serves operational needs and another to serve informational or analytical needs. To some extent, this more enlightened notion of database is due to the advent of PC's, 4gl technology, and the empowerment of the end user.

The split of operational and informational databases occurs for many reasons.

- The data serving operational needs is physically different data from those serving informational or analytic needs.
- The supporting technology for operational processing is fundamentally different from the technology used to support informational or analytical needs.
- The user community of operational data is different from the one served by informational or analytical data.

- The processing characteristics for the operational environment and the informational environment are fundamentally different.

Because of these reasons (and many more!), the modern way to build systems is to separate operational and informational or analytical processing and data.

This book is about the analytical (or the DSS) environment and the structuring of data in that environment. The focus of the book is on what is termed the "data warehouse (or "information warehouse")," which is at the heart of informational, DSS processing.

The discussions in this book are geared toward the manager and the developer. Where appropriate, some level of discussion will be at the technical level. But for the most part, the book is about issues and technique. This book is meant to serve as a guideline for the designer and the developer.

What is analytical, informational processing? It is processing that serves the needs of management in the decision-making process. Often known as DSS (decision support systems) processing, analytical processing looks across broad vistas of data to detect trends. Instead of looking at one or two records of data (as is the case in operational processing), when the DSS analyst does analytical processing, many records are accessed.

In addition, it is very rare for the DSS analyst to update data. In operational systems, data is constantly being updated at the individual record level. In analytical processing, records are constantly being accessed, and their contents are gathered for analysis, but little or no alteration of individual records occurs.

In analytical processing, the response time requirements are greatly relaxed compared to those of traditional operational processing. Analytical response time is measured from 30 minutes to 24 hours. Response times measured in this range for operational processing would be an unmitigated disaster.

The network that serves the analytical community is a much smaller one than the one that serves the operational community. Usually there are many fewer users of the analytical network than of the operational network.

Unlike the technology that serves the analytical environment,

operational environment technology must concern itself with data and transaction locking, contention for data, deadlock, etc.

There are then many major differences between the operational environment and the analytical environment. This book is about the analytical, DSS environment and addresses the following issues:

- granularity of data
- partitioning of data
- metadata
- lack of credibility of data
- integration of DSS data
- the time basis of DSS data
- identifying the source of DSS data—the system of record
- migration and methodology

This book is for developers, managers, designers, data administrators, database administrators, and others who are building systems in a modern data processing environment. In addition, students of information processing will find this book useful.

There are many people who have contributed—directly and indirectly—to this effort. The following list only reflects a few contributors.

- Sue Osterfelt, Storage Tek
- Claudia Imhoff, Connect
- John Zachman, independent consultant
- Jim Kerr, DMR
- Ed Young, Prism Solutions
- Mike Schmidt, Prism Solutions
- Jim Ashbrook, Prism Solutions
- Cynthia Schmidt, Prism Solutions
- Peter LaPorte, Prism Solutions
- Edie Conklin, Prism Solutions
- John Viescas, Tandem Corporation
- George Coleman, Teradata Corporation
- Paul Hessinger, independent consultant
- Jeanne Friedman, Praxis, Inc
- Cheryl Estep, Chevron Corporation
- Nancy Ryan, Storage Tek

- Mark Marshall, IBM
- Ruth Chavez, IBM
- Don Cameron, IBM
- Chuck Ballard, IBM
- Tom Kendra, IBM
- Russ Donovan, IBM
- Arnie Barnett, Barnett Data Systems
- Blair Brown, Computer Performance Analyst and Attorney

Contents

1

Evolution of Decision Support Systems

The world of information systems is an "immature" world. One has to be careful in using that word in public because it normally has a negative connotation. But from a historical perspective it is true. When you compare the history of information processing to that of other professions, there is no contest. We are told that the hieroglyphics in Egypt are primarily the work of an accountant declaring how much grain is owed the pharoah. We walk down the streets of Rome on paths and structures laid out by a civil engineer over 2,000 years ago. Other professions have roots back to antiquity.

So, the information processing profession is historically immature because it has existed only since the early 1960s.

One of the manifestations of the information processing profession's youth is the insistence on dwelling on detail. There is the notion that if we get the details right, the end result will somehow take care of itself and we will achieve success. It's like saying that if we know how to lay concrete, how to drill, and how to install nuts and bolts, we don't have to worry about the shape or the use of the bridge we are building. Such an attitude would drive a more mature civil engineer crazy.

The story of the data warehouse begins with an evolution in which the larger forces at work in the industry are considered. The

larger architecture that is being carved out—of which the data warehouse is the focal piece—is best viewed from a broad, not a detailed, perspective.

THE EVOLUTION

It is interesting that DSS processing is at the end of a long and complex evolution but continues to evolve. The origins of DSS processing hark back to the very early days of computers.

Figure 1.1 shows the evolution of processing from the early 1960s up to 1980.

In the early 1960s, the world of computation consisted of creating individual applications that were run on master files. The applications featured reports and programs, usually COBOL. Punched cards were common. The master files were stored on magnetic tape files, which were good for storing a large volume of data cheaply but had the drawback of needing to be accessed sequentially. Indeed, it is often said that in a given pass of a magnetic tape file where 100 percent of the records have to be accessed, only 5 percent or fewer of the records were actually needed. In addition, accessing an entire tape file may take as long as 20 to 30 minutes, depending on the data on the file and the processing that is being done.

Around the mid-1960s, the growth of master files and magnetic tape exploded. Soon there were master files everywhere. And with that growth came huge amounts of redundant data. The proliferation of master files and massive redundancy of data presented some very insidious problems.

- the need to synchronize data upon update
- the complexity of maintenance of programs
- the complexity of developing new programs
- the amount of hardware required to support all the master files

In short order, the problems of master files that were inherent to the medium itself became stifling. It is an interesting conjecture to speculate what the world of information processing would look like if the only medium for storing data had been the magnetic tape.

If there never had been anything to store bulk data on other than magnetic tape files, the world would have never had large,

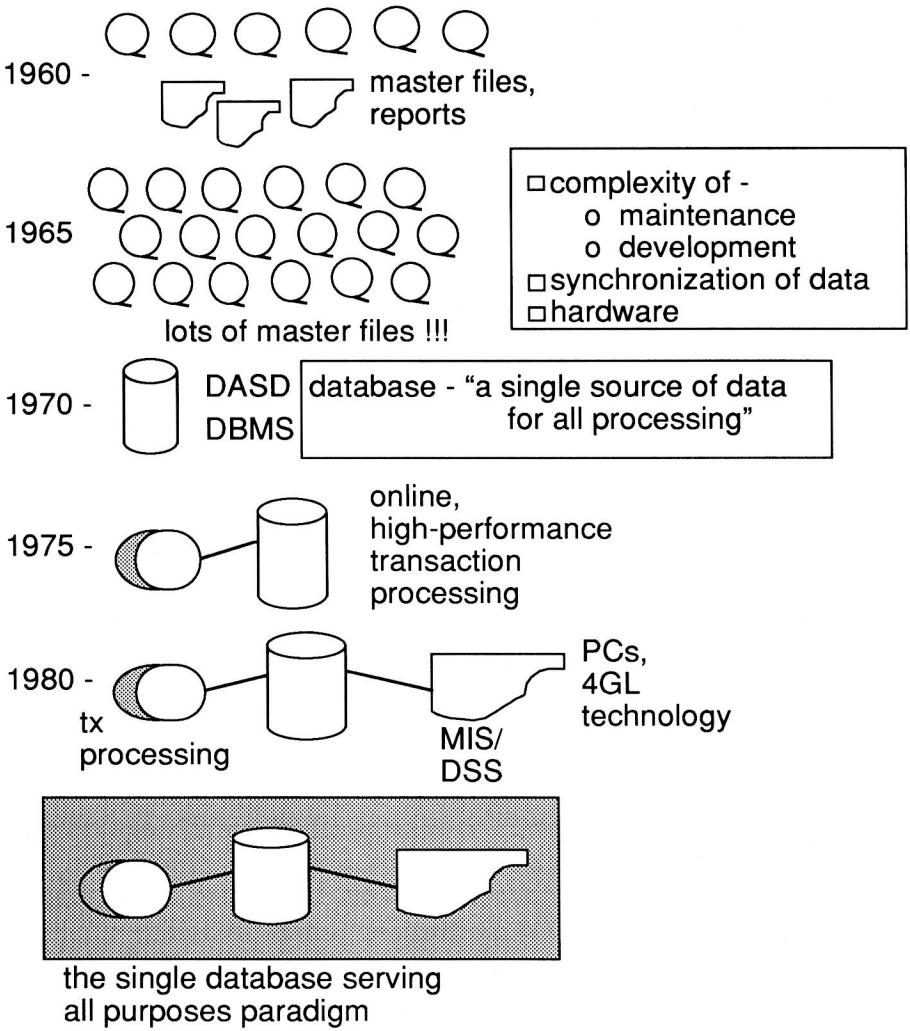


Figure 1.1. The early evolutionary stages of the architected environment.

fast reservations systems, ATM systems, and the like. Indeed, the ability to store and manage data on a variety of media other than magnetic tape files opened up the way for a different and more powerful type of processing that brought the technician and the businessperson together as never before.

THE ADVENT OF DASD

By 1970, the day of a new technology for the storage and access of data had dawned. The 1970s saw the advent of disk storage, or DASD (direct access storage device). Disk storage was fundamentally different from magnetic tape storage in that data could be accessed directly on DASD. There was no need to go through records 1, 2, 3, ... n to get to record n + 1. Once the address of record n + 1 was known, it was a simple matter to go to record n + 1 directly. Furthermore, the time required to go to record n + 1 was significantly less than the time required to scan a tape. In fact, the time to locate a record on DASD was measured in milliseconds.

With DASD came a new piece of system software known as a DBMS or database management system. The purpose of the DBMS was to make it easy for the programmer to store and access data on DASD. In addition, the DBMS took care of such tasks as storing data on DASD, indexing data, and so forth. With DASD and DBMS came a technological solution to the problems of master files. And with DBMS came the notion of a "database." In looking at the mess that was created by master files and the masses of redundant data that aggregated on them, it is no wonder that database is defined as

database—a single source of data for all processing.

But the world did not end in 1970. By the mid-1970s, online transaction processing began to take place on databases. With a terminal and the appropriate software, the technician found that quicker access to data was possible—opening whole new vistas. With high-performance online transaction processing, the computer could be used for tasks not previously possible. The computer could now be used to drive reservations systems, bank teller systems, manufacturing control systems, and the like. Had the world remained in a magnetic tape file state, most of the systems that we take for granted today would not have been possible.

PC/4GL TECHNOLOGY

By the 1980s, more new technologies—such as PC and 4GL technology—began to surface. The end user began to assume a role

previously unfathomed—directly controlling data and systems, outside the domain of the classical data processor. With PCs and 4GL technology came the notion that more could be done with data than servicing online high-performance transaction processing. MIS (management information systems)—as it was called in the early days—could also be done. Today known as DSS, MIS was processing that which was used to drive management decisions. Previously, data and technology were used exclusively to drive detailed operational decisions. There arose the notion of a paradigm—a single database that could serve both operational, high-performance transaction processing and DSS, analytical processing at the same time. Figure 1.1 shows the single database paradigm.

ENTER THE EXTRACT PROGRAM

Shortly after the advent of massive online high-performance transactions, an innocuous program called an "extract" program, began to appear shown in Figure 1.2.

The extract program is the simplest of all programs. It rummages through a file or database, uses some criteria for selection, and, upon finding qualified data, transports the data over onto another file or database.

Soon the extract program became very popular. It permeated the information processing environment. There are (at least!) two reasons for its popularity.

- Because it can move data out of the way of high-performance online processing with an extract program, there is no conflict in terms of performance when the data needs to be analyzed en masse.
- When data is moved out of the operational, transaction processing domain with an extract program, there is a shift in control of the data. The end user ends up "owning" the data once he/she takes control of it.

For these (and probably a host of other) reasons, extract processing was soon found everywhere. By the 1990s, there were many, many extract programs, as depicted in Figure 1.3.

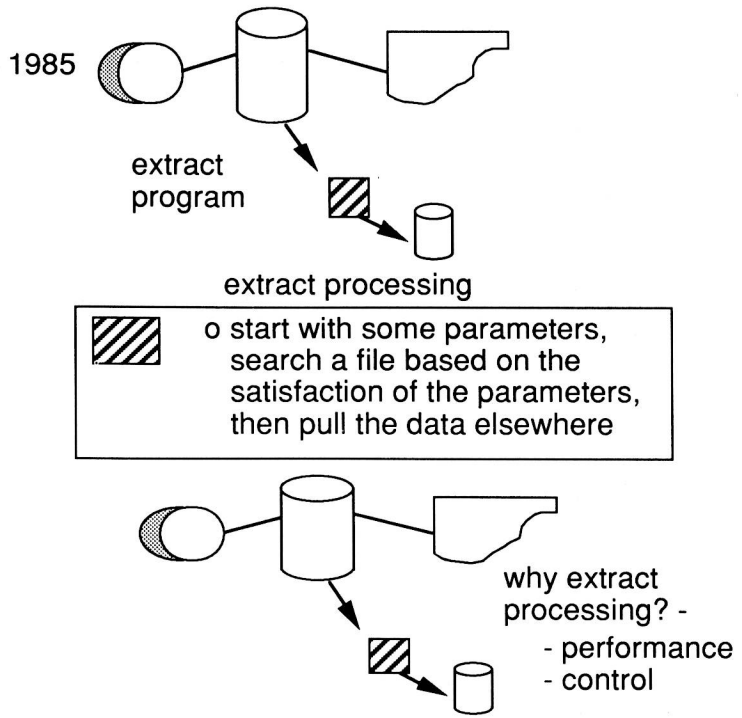


Figure 1.2. Soon there was extract processing.

THE "SPIDER WEB"

Figure 1.3 shows that a "spider web" of extract processing began to form. First, there were extracts. Then there were extracts of extracts. Then extracts of extracts of extracts, and so forth. It was not unheard of for a large company to be doing as many as 45,000 extracts per day.

This pattern of extract processing across the organization became so commonplace that a name developed for it. Extract processing gone out of control produced what can be called the "naturally evolving architecture"—what occurs when an organization handles the whole process of hardware and software architecture with a laissez faire attitude. The larger and more mature the