



Previews of TDWI course books are provided as an opportunity to see the quality of our material and help you to select the courses that best fit your needs. The previews can not be printed.

TDWI strives to provide course books that are content-rich and that serve as useful reference documents after a class has ended.

This preview shows selected pages that are representative of the entire course book. The pages shown are not consecutive. The page numbers as they appear in the actual course material are shown at the bottom of each page. All table-of-contents pages are included to illustrate all of the topics covered by a course.



TDWI Data Modeling

Data Analysis and Design for BI and Data Warehousing Systems

All rights reserved. No part of this document may be reproduced in any form, or by any means, without written permission from The Data Warehousing Institute.

TABLE OF CONTENTS

Module 1	<i>Data Modeling Concepts</i>	<i>1-1</i>
Module 2	<i>Business Data Models</i>	<i>2-1</i>
Module 3	<i>Logical Data Models</i>	<i>3-1</i>
Module 4	<i>Implementation Data Models</i>	<i>4-1</i>
Module 5	<i>Summary and Conclusion</i>	<i>5-1</i>
Appendix A	<i>Entity-Relationship Modeling Basics</i>	<i>A-1</i>
Appendix B	<i>TDWICo Case Study</i>	<i>B-1</i>
Appendix C	<i>Exercises</i>	<i>C-1</i>
Appendix D	<i>Bibliography and References</i>	<i>D-1</i>



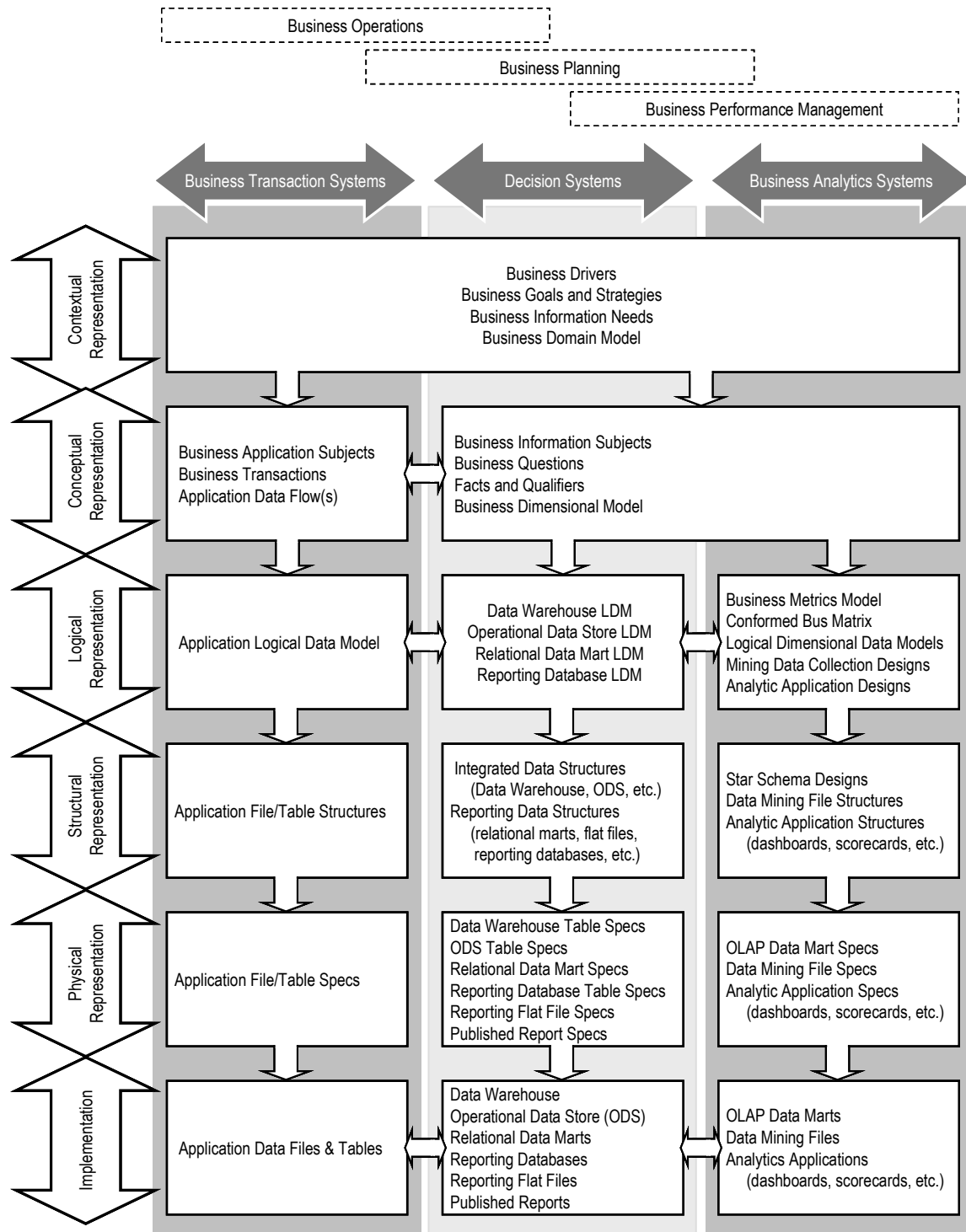
Module 1

Data Modeling Concepts

Topic	Page
The Data Modeling Life Cycle	1-2
Kinds of Data Systems	1-6
Data Characteristics	1-8
Data Modeling Framework for BI	1-10

Data Modeling Framework for BI

Where and What to Model



Data Modeling Framework for BI

Where and What to Model

SCOPE OF DATA ANALYSIS AND DESIGN

The diagram on the facing page illustrates the full scope of data analysis and design as covered in this course. Looking through the diagram you'll find:

- *Business perspectives* of business operations, business planning, and performance management – each dependent on data and information services. Business operations depend primarily on transaction systems with some value received from decision support systems. Business planning is dependent on decision support with some assistance from transaction systems and business analytics. Performance management is primarily an analytics-dependent activity with some application of decision support systems.
- *Six layers of abstraction* from contextual representation of data to implementation of data. The top two layers – contextual and conceptual – represent analysis activities. The third and fourth layers – logical and structural – are design activities. The bottom two layers are directly related to implementation of data systems.
- *Three parallel columns of data analysis and design results* that are based on three distinct kinds of data and information systems – business transaction systems, decision systems, and business analytics systems.

At the contextual level the results are identical for all three columns. It would make little sense if the three kinds of systems were each built using unrelated business context.

At the conceptual level decision support and business analytics systems share common deliverables because they are both founded on enterprise perspective and integrated data. Business transaction systems are likely to be conceptually narrower resulting in non-integrated data and operational systems stovepipes.

Below context and concept levels each type of information system has unique analysis and design deliverables and specialized data models.

- At the center of the model the *Integrated Enterprise Logical Data Model* is highlighted. This model, whether physically created or not, provides an essential business-oriented and application-independent view of the entire scope of data.



Module 2

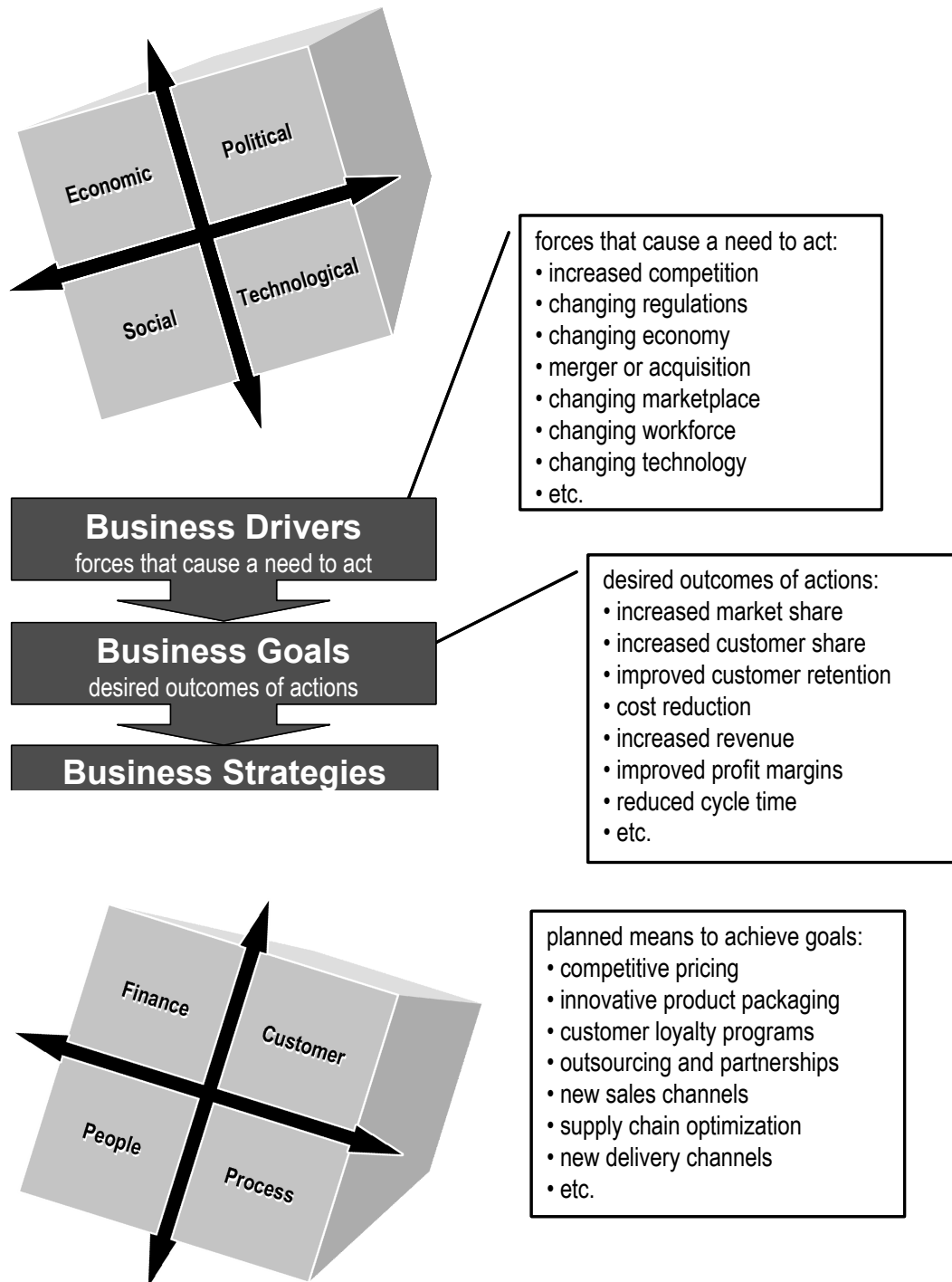
Business Data Models

Topic	Page
Business Context	2-2
Gathering Business Questions	2-10
Analyzing Business Questions	2-18
Fact Analysis and Refinement	2-28
Qualifier Analysis and Refinement	2-32
Fact/Qualifier Analysis Results	2-36
Business Dimensional Modeling	2-38

This page intentionally left blank.

Business Context

Business Drivers, Goals, and Strategies



Business Context

Business Drivers, Goals, and Strategies

THE MODELING FRAMEWORK



WHY MODEL

Business context determines the nature of data and information services – the business processes to be affected, the kinds of applications to be implemented, and the information services to provide. Business context provides the means to align data with business goals.

BUSINESS DRIVERS

Business drivers are those things that are strategically important in positioning the business to achieve its short- and long-term goals. They are the external forces that have significant influence on operation and performance of a business. Drivers create need to take action, but they don't dictate the actions to be taken. Common business driver examples include changing economy, changing marketplace, and changing regulations.

BUSINESS GOALS

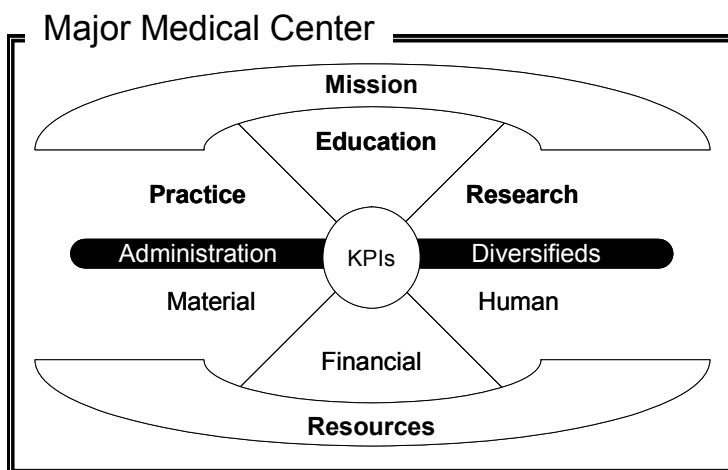
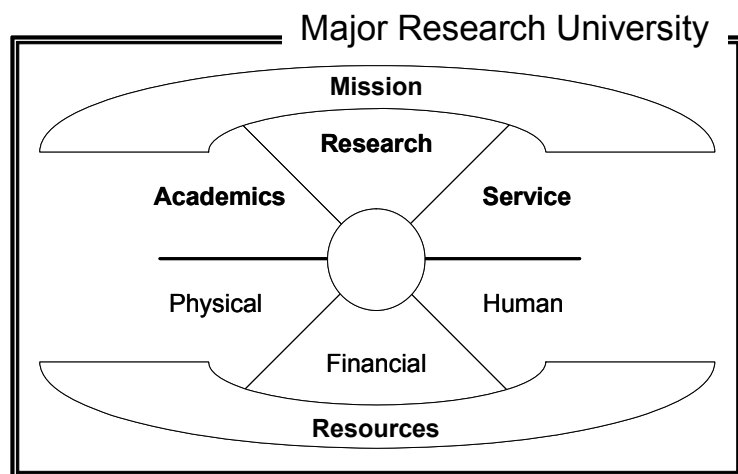
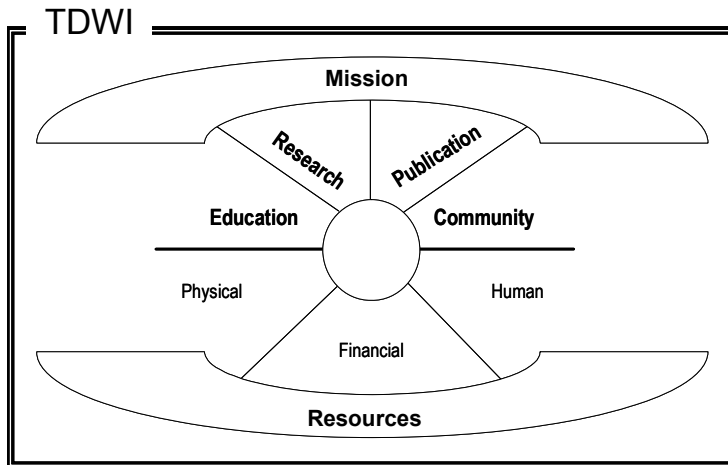
Business goals are the things that the business wants to accomplish to respond to business drivers. Drivers create the need to act. Goals describe the desired outcomes of taking action. Goals are commonly related to financial or operational performance (i.e., cost reduction, generation of revenue, increased market share, etc.) Goals are most effective in setting data management priorities and directions when they are: (1) described by clear, concise, understandable statement, (2) specific enough that level of achievement can be measured, and (3) of high business priority.

BUSINESS STRATEGIES

Business strategies are action plans for the business. They describe how the business plans to accomplish its goals. The range of strategies is broad – introducing new products, exploiting new sales channels, pricing competitively, optimizing business processes, etc. Strategies help to determine which business processes and organizations most need to be information enabled.

Business Context

Business Domains



Business Context

Business Domains

THE MODELING FRAMEWORK



WHY MODEL

A business domain is a sphere of business activity or function – a broad classification of resource and activity that is planned, managed, executed, and monitored by the business. Domain is the top-tier of business data classification (even more broad than subject, for those familiar with subject modeling). Modeling business domains extends the understanding of business context that is established by modeling drivers, goals, and strategies. It is particularly useful as a starting place to model subjects and metrics that align well with the business; and it is valuable throughout the modeling process to prevent losing sight of the “big picture.”

WHAT TO MODEL

A typical enterprise has five to seven domains, each related to the mission of the enterprise, the resources used to fulfill the mission, or the allocation of resources to mission objectives.

The illustration on the facing page shows a nonspecific domain model. The arc across the top represents mission-related domains, each of which is a major category of products or services that the enterprise provides to its customers. The lower arc represents resource-related domains. Resource domains are commonly segmented as human, financial, and physical resources, although the particular names may be different and some enterprises may have unique resource domains (i.e., intellectual).



Module 3

Logical Data Models

Topic	Page
What to Model	3-2
Understanding Data Sources	3-4
Logical Relational Modeling	3-10
Logical Dimensional Modeling	3-20
Logical Models and Business Metrics	3-36
Logical Models and Business Analytics	3-42
Logical Models and Master Data Management	3-46
Logical Models and Unstructured Data	3-50

This page intentionally left blank.

Logical Relational Modeling

The Modeling Process

E-R MODELING

This is the process of developing entity-relationship (E-R) models at the logical level. Remember that logical models are system models that are product independent. They are technology oriented designs, although they are platform-independent. Modeling is an iterative process of identifying entities, attributes, and relationships and refining the model using processes of abstraction, generalization/specialization, and normalization.

ABSTRACTION

Abstraction is the process of formulating general concepts by finding common properties of instances – by looking at what is similar among entities or attributes in the model. Consider, for example, a data model that has an entity *employee* with attributes of *office-phone*, *home-phone*, and *cell-phone*. These attributes are all telephone numbers which could be abstracted as *phone-number* and *phone-number-type*. The resulting of the abstraction is a more flexible model – the ability, for example, to add more phone number types such as fax and emergency contact. Abstraction is a design decision that involves trade-offs. In the example the trade is to gain flexibility at the cost of a more complex data model.

GENERALIZATION & SPECIALIZATION

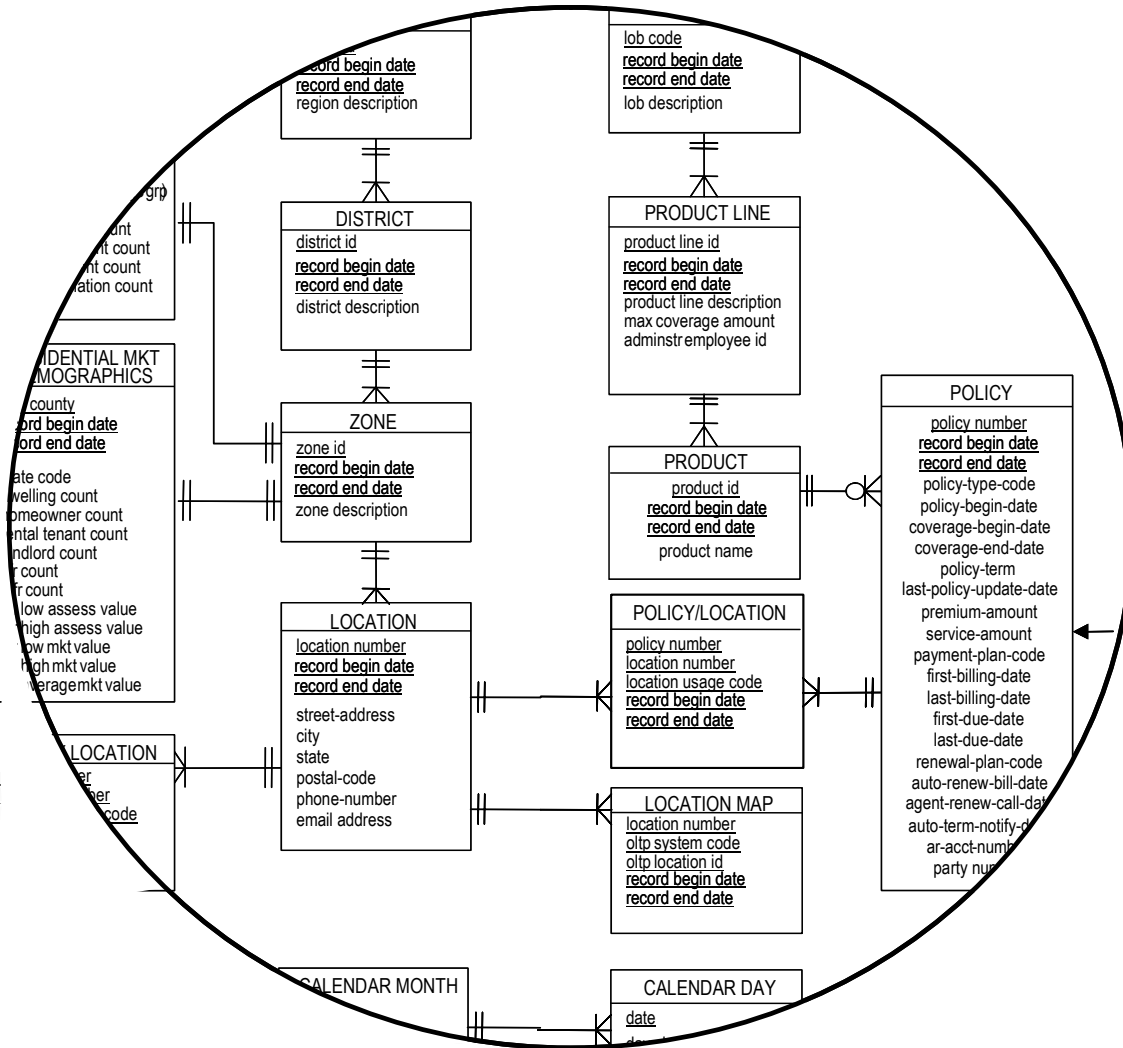
Generalization and specialization are similar to abstraction in that they focus on similarities and differences between entities. An entity type is generalized in response to similarities – creating a *person* entity type, for example, to include the attributes that are common employees, customers, and all other persons in the scope of data design. An entity type is specialized in response to significant differences among occurrences of the entity. *Employee*, for example, might be specialized as *salaried employee* and *hourly employee* because the two types of employees may have different attributes and participate in different relationships.

NORMALIZATION

Normalization is a systematic way of removing redundancies and functional dependencies from a database structure. Normalization is especially important in transaction system databases where redundancy and dependency lead to possible update anomalies and data quality problems. Database theory identifies six normal forms. Common practice generally applies the theory at the 3rd normal form.

Logical Relational Modeling

Logical Models for Data Warehouse and ODS

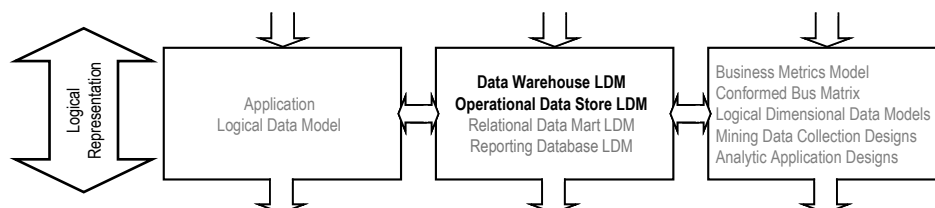


**May Violate 3rd Normal Form to Store Derived Data
Richly Attributed to Meet Unspecified Data Needs
Integrated and Time-Variant View of the Business**

Logical Relational Modeling

Logical Models for Data Warehouse and ODS

THE MODELING FRAMEWORK



WHY MODEL

Both the data warehouse and the ODS need to be supported with a business view that is process independent. Both have integration as a goal and, as already discussed, integration is difficult to achieve without process independence. Furthermore, business users need to understand the contents of the data warehouse and may need to understand ODS contents depending on its defined role. Business understanding is more readily facilitated by business models than by system and technical models.

WHAT TO MODEL

This modeling activity produces an entity-relationship model that represents a business view of the data contained in an integrated data store – a data warehouse or operational data store (ODS).

The logical data model of an ODS is typically fully normalized (or at minimum to the third normal form) and is ideally a true subset of the enterprise model. The data warehouse logical data model is likely to be a combination of an enterprise model subset and extension of that subset. Extensions of the model occur to support needs such as:

- derived data (a violation of the third normal form), and
- aggregate data structures such as master dimension tables (a violation of the second normal form).



Module 4

Implementation Data Models

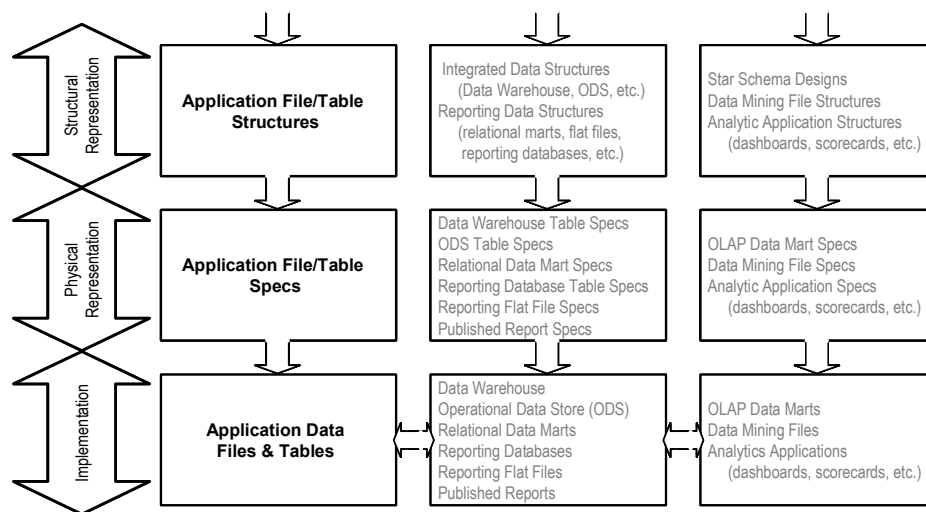
Topic	Page
Data Structure in Transaction Systems	6-2
Structural Modeling and Data Integration	6-4
Normalization	6-6
Time-Variant Data Structures	6-12
Access, Navigation, Security, and Distribution	6-20
Structural Modeling and Business Analytics	6-26
Star-Schema Design	6-28
Analytic Application Data Structures	6-38
Data Mining Data Structures	6-40

This page intentionally left blank.

Data Structure in Transaction Systems

Extracting the Structure of Existing Data

File / Table	Field / Column	Attribute	ID	Entity	Relationship
APMS PREMIUM	<u>Policy_Number</u>	Unique policy identifier	yes	Policy	
	Name	Name of Policyholder		Customer	
	Address	Address of Policyholder		Customer	
	<u>Premium_Amount</u>	Cost of Policy Premium		Policy	
	<u>Policy_Term</u>	Coverage duration		Policy	
	<u>Begin_Date</u>	First date of coverage		Policy	
	<u>End_Date</u>	Last date of coverage		Policy	
	<u>Discount_Code</u>	Identifies kind of discount	partial	Discount	DISCOUNT → POLICY
APMS POLICY	<u>Policy_Number</u>	Unique policy identifier	yes	Policy	
	<u>Customer_Number</u>	Unique customer identifier	yes	Customer	CUSTOMER → POLICY
	VIN	Vehicle identification number	yes	Vehicle	VEHICLE → POLICY
	Make	Vehicle manufacturer		Vehicle	



Data Structure in Transaction Systems

Extracting the Structure of Existing Data

LEVELS OF TRANSACTION SYSTEM MODELS

Four levels of transaction systems data and models are of interest for the value that they contribute to data warehouse modeling:

- *Implemented Data* already exists as application data files and tables.
- *Physical models* usually exist in part as descriptions of data structures (DDL, COBOL copy code, etc.). Multiple, potentially confusing, descriptions may exist for a single file or table. This is especially true of older legacy systems.
- *Structural Models* may need to be created to identify and resolve conflict where multiple physical descriptions exist for a file or table.
- *Logical Models* exist infrequently. Where logical models are present, they are often incomplete or out of date, and are almost certainly limited to a single application's view of the data..

BOTTOM-UP DATA SOURCE MODELING

It is valuable to build a source data model in reverse – working from implemented data backward E/R model with. The process is one of examining every data element (column or field) in every data store (file or table) to answer the questions:

- What business fact (attribute) does this data element contain?
- What thing (entity) is it a fact about?
- Does the data element identify the entity that it describes?
- Does the data element indicate a relationship to another entity?

The answers in a spreadsheet format produce a table such as that below which is readily translated into an E/R diagram if necessary.

File / Table	Field / Column	Attribute	ID	Entity	Relationship
APMS PREMIUM	Policy_Number	Unique policy identifier	yes	Policy	
	Name	Name of Policyholder		Customer	
	Address	Address of Policyholder		Customer	
	Premium_Amount	Cost of Policy Premium		Policy	
	Policy_Term	Coverage duration		Policy	
	Begin_Date	First date of coverage		Policy	
	End_Date	Last date of coverage		Policy	
	Discount_Code	Identifies kind of discount	partial	Discount	DISCOUNT → POLICY
APMS POLICY	Policy_Number	Unique policy identifier	yes	Policy	
	Customer_Number	Unique customer identifier	yes	Customer	CUSTOMER → POLICY
	VIN	Vehicle identification number	yes	Vehicle	VEHICLE → POLICY
	Make	Vehicle manufacturer		Vehicle	



Module 5

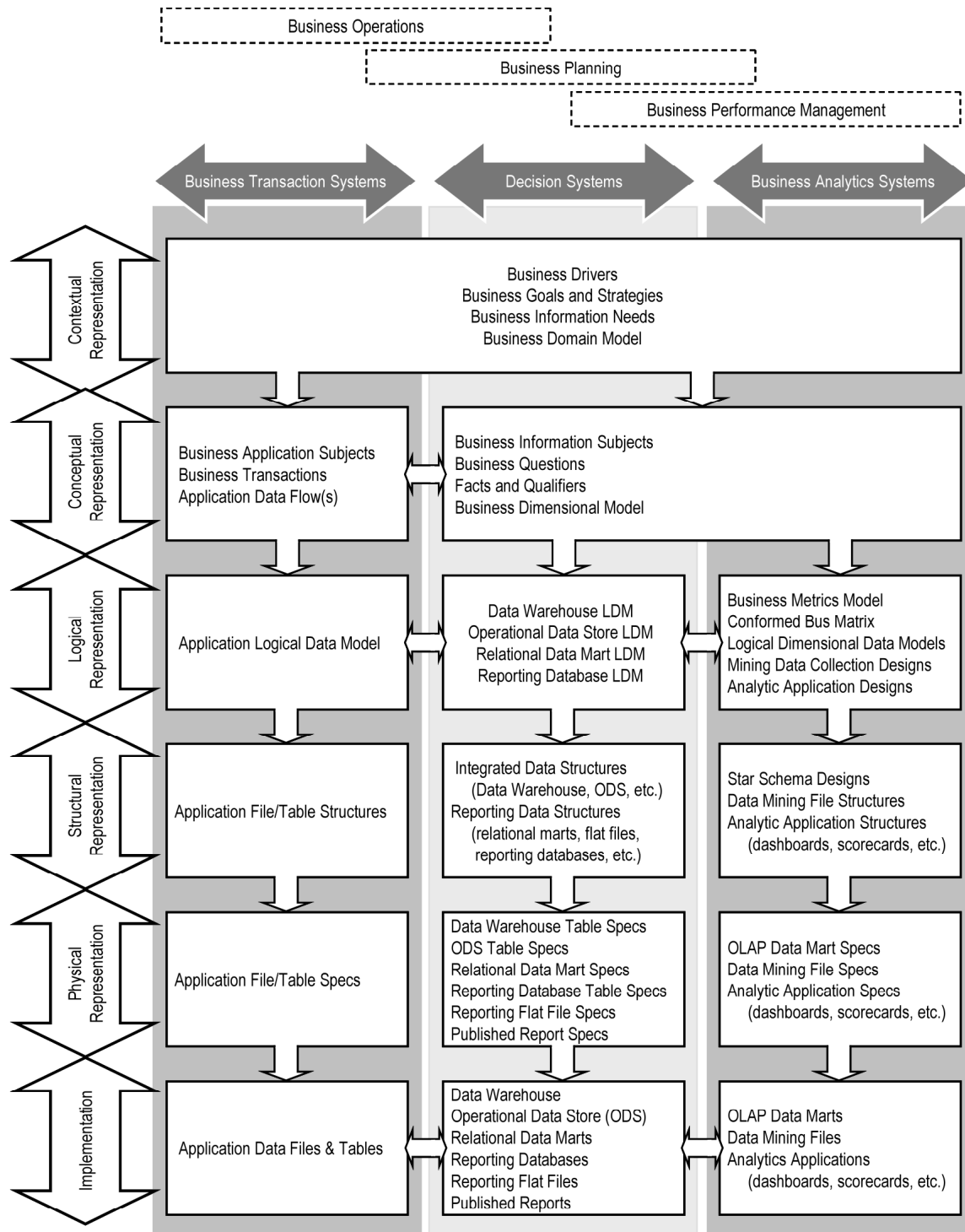
Summary and Conclusion

Topic	Page
A Quick Review	5-2

This page intentionally left blank.

A Quick Review

The Data Modeling Landscape



A Quick Review

The Data Modeling Landscape

MANY LEVELS, MANY MODELS, MANY METHODS

The range of data modeling possibilities in data warehousing and business intelligence is significantly larger than for modeling of transactional and operational systems. You may need to model across multiple levels of abstraction, to produce distinct models for warehouses and data marts, and to employ both relational and dimensional modeling techniques.

The diagram on the facing page illustrates a robust picture of the possibilities. It does not suggest that any data warehousing project – or even every warehousing program – must include all of these models.

The key to success in warehouse data modeling is to be familiar with all of the possibilities – to fill your modeling toolbox with all of the tools – then choose the right tool for each modeling need.