

# Data Modeling Master Class

## *Steve Hoberman's Best Practices Approach to Developing a Competency in Data Modeling*

Data modeling is about understanding the data used within our operational and analytics processes, documenting this knowledge in a precise form called the “data model”, and then validating this knowledge through communications with both business and IT stakeholders. Underlying all successful applications is a robust and precise data model, and similarly, most software development failures are due to a lack of understanding of the data or data requirements.

A data model is therefore an essential part of applications development including forward engineering, reverse engineering, and integration efforts. Forward engineering means focusing on business requirements, whereas reverse engineering means modeling existing systems to drive the support, replacement, or customization of applications. Integration projects such as business intelligence efforts, data lakes, and master data initiatives, require a consistent holistic view of concepts such as Customer, Account, and Product.

The Master Class is a complete data modeling course, containing three days of practical techniques for producing conceptual, logical, and physical relational and dimensional and NoSQL data models. After learning the styles and steps in capturing and modeling requirements, you will apply a **best practices approach** to building and validating data models through the Data Model Scorecard®. You will know not just how to build a data model, but how to build a data model *well*. Two case studies and many exercises reinforce the material and will enable you to apply these techniques in your current projects.

### Top 5 Objectives

1. Determine how and when to use each data modeling component.
2. Apply techniques to elicit data requirements as a prerequisite to building a data model.
3. Build relational and dimensional conceptual, logical, and physical data models.
4. Assess the quality of a data model.
5. Incorporate supportability and extensibility features into the data model.

## Prerequisite(s)

This course assumes no prior data modeling knowledge and, therefore, there are no pre-requisites. This course is designed for anyone with one or more of these terms in their job title: “data”, “analyst”, “architect”, “developer”, “database”, and “modeler”.

## Topics

### Part 1: Modeling Basics

Assuming no prior knowledge of data modeling, we introduce our first case study which illustrates four important gaps filled by data models. Next, we will explain data modeling concepts and terminology, and provide you with a set of questions you can ask to quickly and precisely build a data model. We will also explore each component on a data model and practice reading business rules. We will complete several exercises, including one on creating a data model based upon an existing set of data. You will be able to answer the following questions by the end of this section:

- What is a data model and what characteristic makes the data model an essential wayfinding tool?
- How does the 80/20 rule apply to data modeling?
- What three critical skills must the data modeler possess?
- What six questions must be asked to translate ambiguity into precision?
- Why is precision so important?
- What three situations can ruin a data model’s credibility?
- What are three key skills every data modeler should possess?
- Why are there at least 144 ways to model any situation?
- What do a data model and a camera have in common?
- What are the most important questions to ask when reviewing a data model?
- What are entities, attributes, and relationships?
- Why subtype and how do exclusive and non-exclusive subtypes differ?
- How do different modeling notations represent subtypes?
- What are candidate, primary, natural, alternate, and foreign keys?
- What are the perceived and actual benefits of surrogate keys?
- What is cardinality and referential integrity and how do they improve data quality?
- How do you “read” a data model?
- What are the different ways to model hierarchies and networks?
- What is recursion and why is it such an emotional topic?

## Part 2: Overview to the Data Model Scorecard®

The Scorecard is a set of ten categories for validating a data model. We will explore best practices from the perspectives of both the modeler and reviewer, and you will be provided with a template to use on your current projects. Each of these following ten categories heavily impacts the usefulness and longevity of the model:

- Ensuring the model captures the requirements
- Validating model scope
- Understanding conceptual, logical, and physical data models
- Following acceptable modeling principles
- Determining the optimal use of generic concepts
- Applying consistent naming standards
- Arranging the model for maximum understanding
- Writing clear, correct and consistent definitions
- Fitting the model within an enterprise architecture
- Comparing the metadata with the data

## Part 3: Ensuring the model captures the requirements

There is no one way to elicit requirements – rather it requires knowing *when* to use certain elicitation techniques such as interviewing and prototyping. We will focus on techniques to ensure the data model meets the business requirements. You will be able to answer the following questions by the end of this section:

- What is the *Requirements Lifecycle*?
- Why do we “elicit” instead of “gather” requirements?
- When should you use closed questions vs. open questions during an interview?
- How do you perform data archeology during artifact analysis?
- What are two creative prototyping techniques for the non-techie?
- How can you validate that a data model captures the requirements without showing the data model?

## Part 4: Validating model scope

We will focus on techniques for validating that the scope of the requirements matches the scope of the model. If the scope of the model is greater than the requirements, we have a situation known as “scope creep.” If the model scope is less than the requirements, we will be leaving information out of the resulting application. You will be able to answer the following questions by the end of this section:

- How do you define “metadata” in practical terms?
- What techniques can you use to avoid scope creep?
- When is observation (job shadowing) an effective way to capture requirements?
- What are the different techniques for initiating an interview?
- What are the three job shadow variations?
- How can prototyping assist with defining model scope?

## Part 5: Understanding conceptual, logical, and physical data models

The conceptual data model captures a business need within a well-defined scope, the logical data model captures the business solution, and the physical data model captures the technical solution. Relational, dimensional, and NoSQL techniques will be described at each of these three levels. We will also practice building several data models and you will be able to answer the following questions by the end of this section:

- How do relational and dimensional models differ?
- What are the ten different types of data models?
- What are the five strategic conceptual modeling questions?
- Why are conceptual and logical data models so important?
- What are the Concept and Question Templates?
- What are four different ways of communicating the conceptual?
- What are six conceptual data modeling challenges?
- What are the five steps to building a conceptual data model?
- What is the difference between grain, base, and atomic on a dimensional?
- What are the three different paths for navigation on a dimensional data model?
- What are the differences between transaction, snapshot and accumulating facts?
- What are the three different variations of conformed dimensions?
- What are junk, degenerate, and behavioral dimensions?
- What are outriggers, measureless meters, and bridge tables?
- What are some dimensional modeling do’s and don’ts?
- How can you leverage the grain matrix to capture a precise and program-level view of business questions?
- What is the difference between a star schema and a snowflake?
- What is normalization and how do you apply the Normalization Hike?
- What is the Attributes Template?
- Where should denormalization be performed on your models?
- What are the five denormalization techniques?
- What is the difference between aggregation and summarization?
- What are the three ways of resolving subtyping on the physical data model?

- What are views, indexing, and partitioning and how can they be leveraged to improve performance?
- What are the four different types of Slowly Changing Dimensions?
- What is the lure of NoSQL?
- What are the four characteristics NoSQL differs from RDBMS?
- What are Document, Column, Key-value, and Graph databases?
- What are the advantages and disadvantages of going “schema-less”?
- What is the difference between ACID and BASE?
- What is MongoDB and is there a difference between a physical and implementation data model?

## Part 6: Following acceptable modeling principles

We will cover *Consistency*, *Integrity*, and *Core* modeling principles. You will be able to answer the following questions by the end of this section:

- What tools exist to automate checking model structure?
- What are circular relationships and why are they evil?
- Why are good default formats really bad?
- What are the most common structural violations on a data model?
- Why should you avoid redundant indexes?
- Why shouldn't an alternate key be null?
- How do you catch definition inconsistencies?
- What is a partial key relationship?
- Why must a subtype have the same primary key as its supertype?

## Part 7: Determining the optimal use of generic concepts

Abstraction is a technique for redefining business terms into more generic concepts such as Party and Event. This module will explain abstraction and cover where it is most useful. You will be able to answer the following questions by the end of this section:

- What is abstraction and at what point in the modeling process should it be applied?
- What three questions (known as the “Abstraction Safety Guide”) must be asked prior to abstracting?
- What is the high cost of having flexible structures?
- How does abstraction compare to normalization?
- What are the three levels of data model patterns?
- Why are roles so important to analytics?
- What are metadata entities?

- Why does context play a role in distinguishing event-independent from event-dependent roles?
- What are industry data models and where do you find them?

## **Part 8: Applying consistent naming standards**

Consistent naming standards will get your organization one step closer to a successful enterprise architecture. We will focus on techniques for applying naming standards and you will be able to answer the following questions by the end of this section:

- What is naming structure, term, and style and how do they apply to entities, attributes, and relationships?
- What are the three most important parts of a naming standards document?
- What is a Reference Guide?
- Why is an “enforcer” required for standards compliance?
- What is the ISO 11179 standard and how can it help my organization?

## **Part 9: Arranging the model for maximum understanding**

A data model is a communication tool and if the model is difficult to read it can hamper communication. We will focus on techniques for arranging the entities, attributes, and relationships to maximize readability. You will be able to answer the following questions by the end of this section:

- How can our modeling tools make readability an easy category to ace?
- Why is keeping relationship lines as short as possible better than minimizing crossing lines?
- Why should we not alphabetize attribute names?
- Why should we avoid UPPERCASE?
- Why should we organize attributes in a transaction entity by classword, and attributes in a reference entity by chronology?

## **Part 10: Writing clear, complete, and correct definitions**

Although definitions may not appear on the data model diagram itself, the definitions are integral to data model precision. We will focus on techniques for writing useable definitions. You will be able to answer the following questions by the end of this section:

- How do you play Definition Bingo?
- Why are definitions so much more important now than they were in the past?
- What are best practices for writing a good definition?

- How do you validate a definition?
- How do you reconcile competing definitions?
- What is the Consensus Diamond and how can being aware of Context, State, Time, and Motive improve the quality of our definitions?
- What are some workarounds when you cannot get common agreement on a definition (e.g. the Batman technique)?

## Part 11: Fitting the model within an enterprise architecture

A data modeler is not only responsible to the project for capturing the application requirements, but also responsible to the organization to ensure all terms and relationships are consistent within the larger framework of the enterprise data model. We will focus on techniques for ensuring the data model fits within a “big picture”. You will be able to answer the following questions by the end of this section:

- What is the Data Vault and how do you build a Data Vault using hubs, links, and satellites?
- What is an enterprise data model and why have one?
- What are the secrets to achieving a successful enterprise data model?
- Why is enterprise *mapping* more important than enterprise *modeling*?
- What three program initiatives benefit most from an enterprise data model?

## Part 12: Comparing the metadata with the data

A logical or physical data model should not be considered complete until at least some data analysis has been done on the data that will be loaded into the resulting data structures. We will focus on techniques for confirming that the attributes and their rules match reality. Does the attribute **Customer Last Name** really contain the customer’s last name, for example? You will be able to answer the following questions by the end of this section:

- How can domains help improve data quality?
- What are the three main types of domains?
- How can I capture lineage using the Family Tree?
- Why is the Family Tree an important reality check?
- How can the Data Quality Validation Template help us with catching data surprises early?

## Quotes on the Data Modeling Master Class

*Thanks again for such an excellent opportunity to participate in this master class, it was extraordinary! The coverage of data modeling, the pace, the content, the interaction with you and the class, all of it was just awesome!*

- S. Johnson, U.S. Dept. of Energy

*This was the most comprehensive, informative, energetic, interesting and just plain FUN class I have ever taken on the subject of Data Modeling.*

- G. Werner, Long Island Railroad

*In my long professional career, I have participated in many training seminars but I have never encountered a class in which the subject had been so thoroughly considered and presented in such a clear and engaging manor. As a fairly new, but full time data modeler, I expect to use the things that I learned in this class every day.*

- G. Schmid, Travelers Insurance

*An excellent course with focus on the basics. I can only imagine the effort that went into developing this course. Lucky that I happened to find it.*

- R. Sampath, Deloitte

*Having never done data modeling before, I can now say, I am excited about implementing the skills I've learned.*

- L. Felder, Johns Hopkins HealthCare

*Truly enjoyed this class even though I have been modeling databases for 24 years.*

- M. Austin, Wells Fargo

*Every participant was at a different level of knowledge and everyone learned from the class.*

- S. Slivova, Senior Business Analyst, Gen Re - A Berkshire Hathaway Company

## Speaker

Steve Hoberman has trained more than 10,000 people in data modeling since 1992. Steve is known for his entertaining and interactive teaching style (watch out for flying candy!), and organizations around the globe have brought Steve in to teach his **Data Modeling Master Class**, which is recognized as the most comprehensive data modeling course in the industry. Steve is the author of nine books on data modeling, including the bestseller *Data Modeling Made Simple*. Steve is also the author of *Blockchainopoly*. One of Steve's frequent data modeling consulting assignments is to review data models using his Data Model Scorecard® technique. He is the founder of the Design Challenges group, Conference Chair of the Data Modeling Zone conferences, director of Technics Publications, and recipient of the Data Administration Management Association (DAMA) International Professional Achievement Award.