Getting the “Business” into Business - Oriented Data Modeling – Practical Techniques & Examples

www.ERwin.com
Table of Contents

Data Modeling: Not Only a Technical Endeavor .................................................................................................................. 3
Business-Oriented Data Modeling: Restraining Factors ........................................................................................................... 4
Techniques and Attitudes for Successful Business-Oriented Modeling .......................................................................................... 5
Case Study 1 – Data Models Help Regulatory Agency Launch New Program and Processes ................................................. 7
Case Study 2 – Global Technology Company Improves Marketing & Sales Processes with Data Models ......................... 8
Case Study 3 – HR Organization Determines Future of Business Software through Data Modeling .............................. 9
Conclusion: Common Success Factors ........................................................................................................................................ 11
Introduction

Data modeling is often seen as a technical discipline used only by data professionals for database design. In fact, it is often seen as the *same thing* as database design. However, with the right perspective, techniques, and tools, data models have also proven useful to non-technical stakeholders. Business leaders and subject matter experts find that well-structured conceptual models provide a new way to see and understand their business and make business policy decisions. Business analysts use data models to discover requirements that would otherwise be missed entirely or not discovered until much later. Three keys to making data models relevant for these business professionals are:

- Knowing how to focus on the business before focusing on business data;
- Understanding the differences between conceptual, logical, and physical models, and how they each serve different needs and audiences;
- Having specific techniques and supporting tools that are appropriate for non-technical stakeholders.

This white paper illustrates these points and provides specific guidelines and techniques for a business-oriented approach to data modeling. Examples demonstrate how business professionals have appreciated and used data models, and how these models have had a positive effect on their business.

Data Modeling: Not Only a Technical Endeavor

Data modeling is not only a technical endeavor, and it was a learning process for me in my career to come to this realization. Sometime around 1980, I built my first “logical data model.” It looked a lot like an IMS hierarchical database, which isn’t surprising - at the time I was an IMS DBA just moving into the Data Administration function. My new manager in the Data Administration group wanted the logical data model for a system I’d just worked on, and I assumed that as long as I didn’t show all the arcane technical details like keys and pointer options, that would be good enough. It wasn’t. I just didn’t “get” that a data model could be anything other than the database design. Some thirty years later, that’s still a widely held and problematic assumption outside the data management community.

Eventually, I learned that a logical data model was different from a physical database design, and much later than that I learned that a conceptual model was very different from a logical model. Making that initial physical-logical distinction was a struggle because methods at the time were still very focused on technical design and performance issues. “Business-oriented” approaches to modeling and design were emerging, but weren’t yet widespread and were often still mired technical in detail. I knew there had to be a better way, but I just didn’t know what it was.

The “aha” moment came at a data administration conference in 1983 when a speaker put out the controversial idea that most of our data design methods were not only complex and labor-intensive, they prevented the involvement of the people with the most knowledge - the business people themselves. He claimed that data structures were “directly derivable from the experience of the business people” and described some methods that I later realized drew on high-level Entity-Relationship modeling. On the way to that conference I happened to pick up “In Search of Excellence1” at an airport bookshop. It crystallized my sense that the methods I’d been using were missing the mark - they completely went against core principles from that landmark book, especially “a bias for action,” “close to the customer,” and “productivity through people” (direct involvement of all levels.) What I learned at the conference seemed a lot more “excellent” than what I’d been doing, and I returned home eager to experiment.

By then, I was working as a consultant, and my client was a Data Administration group that was just as eager as I was for change. They encouraged me to put the ideas into action, and provided the opportunity. In the end, the techniques were more successful than we imagined possible, but we started small. Initially, we helped out at the beginning of projects by facilitating sessions to clarify scope, terminology, and basic policies, rules, and requirements. We didn’t talk about “data” or “data modeling”, instead focusing on “things you care about,” “what you need to know about those things,” terms and definitions, “how these things relate to one another,” rules and policies, and so on. We took pains to avoid having it look like

---

a technically oriented technique imposed by the IT function - we were just “discussing your business”. These sessions were very helpful in clarifying scope and requirements, but business participants recognized another benefit - they understood their own business better than they had before because of seeing it in a new way. They were used to seeing themselves in terms of organizational units, job titles, procedures, processes, and so on, but no one had seen a view stripped down to “the things we need to know about.”

Eventually, this success led to enterprise-wide, conceptual modeling engagements that spanned multiple departments and divisions of the company. The term hadn’t been coined yet, but this was “cross-functional” work. These modeling efforts were so successful at improving communication within and across organizations that unsolicited requests for more of the same came in from the business. Imagine that – business professionals asking for data modeling!

The next large organization I consulted for embraced data modeling even more enthusiastically, and created enterprise models for areas such as “Customer” and “Product.” Again, the themes of “better understanding” and “better communication” emerged, as they so often do when business-oriented modeling is practiced. But it wasn’t just a feel-good exercise. Better systems and databases were developed as a result of these coherent, overarching models. The company was well-positioned when data warehousing and business intelligence emerged because they already had an integrated view of their data with consistent naming and clear definitions. The outcome I received the most satisfaction from was that some of the Vice Presidents kept conceptual-level Entity-Relationship Diagrams on their office walls, and actually used them to guide discussions and understand policy decisions. Imagine that - corporate executives with data models on their office walls.

These examples demonstrate that it is indeed possible to have business people fully engaged in developing and using data models, including front-line workers, subject matter experts, and all levels of management. This not only benefited the enterprise in ways that might not be expected from data modeling, such as improved communication, but it also led to very practical benefits as well. Direct business involvement greatly increased the quality and longevity of those data models, and therefore the quality and longevity of the systems and databases using them, leading to long term cost benefits. Unfortunately, this isn’t the case in every organization. In the following section, we’ll look at factors that interfere with the wider use of data modeling, and at specific techniques and guidelines that have helped at many enterprises.

**Business-Oriented Data Modeling: Restraining Factors**

As we’ve demonstrated, data modeling can be a powerful technique, and appropriately used, it is also a fundamental technique, because it focuses on the fundamental things of interest to an enterprise. These things are the entities that:

- The enterprise needs information about
- Business processes operate on
- Applications manipulate
- Business rules reference
- Record-keeping systems (e.g., databases) record information about

When you think about it, it is almost impossible to describe an information need, a business process, an essential piece of functionality (a requirement), or a business rule without reference to an entity or some other component of a data model. In our practice, we’d never consider doing business process change or application requirements work, for instance, without developing at least a conceptual data model. It’s staggering, then, that data models aren’t more widely used to develop a common understanding of things before expressing the information needs, processes, requirements, and so on that reference those things. Why would this be so?

There is a combination of factors that interfere with a business-oriented approach to data modeling and its use outside of technical arenas like database design, but five regularly stand out:

1. The belief that data models are only useful for data problems, such as documenting data requirements or designing data structures. This view has developed slowly over the years, as companies have drifted away from a
business orientation toward the view that data models are the same thing as physical database designs.

2. This has been exacerbated by the widespread use of “data modeling” techniques that are really closer to physical database design, appropriate for technical audiences but inappropriate and irrelevant for business audiences. A common example is the modeler who is uncomfortable identifying a new entity without specifying its primary key, or defining a relationship without knowing what foreign key will eventually support it.

3. A lack of awareness that there are two very different types of data models, each with a particular level of detail and purpose, before getting into physical database design (or physical data models, if you prefer.) These are conceptual and logical data models. As well, there is little clarity and consistency within the data management field on what a conceptual model is, and how it is different than a logical model. At many organizations, conceptual and logical data models are treated as being more or less the same thing, when in fact there are major, specific differences between them.

4. The lack of explicit connections between data modeling and the other techniques a business analyst is likely to employ, such as business process modeling or use cases. As noted, these connections are inherently there, because processes, application functions, business rules, and information needs are all expressed in terms of entities, relationships, and attributes, but this connection has not been made obvious in most of the published methodologies.

5. Finally, even if the previous issues are addressed, data models are often maintained in a specialized tool or repository that is inaccessible to a wider business audience. It’s been demonstrated repeatedly over recent years that if technology can be deployed to make information more readily available, people will use it.

That list is a bleak assessment, but there’s good news. First, simple, easily adopted techniques can make all the difference in making data modeling more business-friendly and useful. Second, tools are becoming available that can help get data models seen and used by a wider audience. Two examples come to mind. The new CA ERwin’s Web Portal makes data models easily accessible via a familiar web interface and CA ERwin’s metadata bridges can leverage data model information by exchanging it with business process management, business intelligence, or requirements management tools. If you can make information easily accessible by either exporting directly into the tools that business workers are using, or providing information in an intuitive web interface that’s easy to access and understand, they are more likely to see data models as an asset that can help them, rather than an administrative burden that they must endure.

Techniques and Attitudes for Successful Business-Oriented Modeling

We’ll now describe some factors that contribute to successful business-oriented modeling, and, even more important, the mindset that supports them. Year in and year out, in organizations of every type, five key factors repeatedly emerge. A brief overview of each will be provided, and in the next section we’ll demonstrate them with three real-life examples.

1 – Mindset: “It’s a description of a business”

A commonly held view is that a data model “is a description of data requirements” or “is the design of a data structure” or something similar. That’s perfectly correct, but only for logical or physical data models, neither of which you really want to show to a business audience. Conceptual data models (or, simply, conceptual models) are the ones that are ideal for business use, and they shouldn’t be seen as data requirements or a data design. A conceptual data model is simply a description of a business. One way to make this point is to think about other types of models. What is a simple business process flow model, such as a swimlane diagram? It’s a description of a business, in terms of what it does. What is an org chart? It’s a description of a business, in terms of how it’s organized. And what is a conceptual data model? It’s a description of a business in terms of the things (the “concepts”) that it deals with. It’s nowhere nearly detailed enough to provide a blueprint for database design, or document all data requirements, but it’s certainly enough to forge agreement and

---

2 In fact, there is a third type, the contextual data model, which is a very high-level view used primarily for scoping purposes; this type of model is beyond the scope of this paper.
understanding of key concepts, whether or not we know yet what the specific data requirements are. (Later, when we concentrate on “data we need about those things”, we are moving into logical modeling.)

To practice this “description of a business” perspective, deliver a presentation of a conceptual model to your colleagues and use only business terminology, not data modeling terminology. Remember, business people intuitively work with models all the time, as when they create a diagram in PowerPoint to illustrate a concept. We see again and again that they intuitively “get” data models, as long as you don’t start with arcane terminology. Once a business audience is comfortable with models, you can begin introducing data modeling terms.

2 – Have a clear understanding of the difference between conceptual and logical models

If you survey a group of data modelers, you find a wide range of opinions on what, exactly, a conceptual model is. Many modelers believe it’s essentially the same as a logical model. However, there are real and important differences - a conceptual model is focused on core business concepts, while a logical model focuses on the details of the structure needed to maintain records about those concepts. A conceptual model is very different from a logical model in the following ways:

- A conceptual model is a business-oriented overview that focuses on the most important things or concepts in a business, while a logical data model focuses on capturing all data requirements and structural rules.
  A logical model is at the level of fully normalized, fully attributed, and fully propertized entities;
- A logical data model will have, on average, five times as many entities as the conceptual model. Why? Because the logical model is fully normalized, while the conceptual model shows only the fundamental “things” of interest to the business. A key guideline – every entity in a conceptual model will be something that the business talks about on a daily basis.
- The preceding points mean that the logical model has components that just won’t be relevant in a conceptual model – primary keys, foreign keys, all of the attributes, attribute properties, relationship optionality, and other technical details.

3 – Use graphic principles on your data model diagrams that support understanding

Earlier we compared data models to org charts and process flow models. Think about how those are invariably drawn – with a sense of direction. Org charts are almost always drawn top-down, as a hierarchy, and process flow models are drawn left-to-right, according to sequence and dependency. We should draw data models exactly the same way, with a sense of direction. That could be bottom to top, left to right, but experiments prove that the vast majority of business people will prefer a data model drawn such that dependency flows from top to bottom. That is, at the top of the diagram will be the independent entities, and below that their children (characteristic and associative entities) and then their children below that, and so on. This makes it easy for the business to focus on the most fundamental things first before working down through layers of detail. By contrast, the vast majority of data models are apparently drawn according to some other principle – no crossed lines, or important entities in the middle, or just putting the entities where they fit. One of my ERwin data modeling blogs, “Top-Down How-To” explored this topic in detail. Figure 1 shows an extract of a simple diagram from that post illustrating a model drawn top-down.

![Figure 1](http://erwin.com/expert_blogs/detail/top-down_how-to/)
4 – Appeal to all learning styles: Visual, Auditory, and Kinesthetic

It’s widely recognized that individuals have a preferred “learning style,” the mode in which they best receive and retain information. These are:

- Visual – diagrams or other schematic depictions
- Auditory – words (narratives) whether written or spoken
- Kinesthetic – tangible examples, such as scenarios

During development of a model, all three of these should be employed. As data modelers, though, our basic mode is visual (the Entity-Relationship Diagram) and modelers often don’t use any other techniques. The most experienced data modelers, however, extensively use auditory modes early in the data modeling process, even before they start diagramming. The kinesthetic perspective can be incorporated by using scenarios (like test cases) or “props” such as report or screen layouts.

Once, at a car insurance company, we struggled with how to depict the relationship among people, policies, and vehicles. Only when we introduced scenarios could we get the necessary information from the business – “John and Beth jointly hold a policy on a 2002 Ford Taurus. John is at fault for an accident while he is driving the vehicle. A penalty (a premium increase) will be applied. Who will pay this penalty?”

Even though each person has a preferred style, everyone benefits when all three are employed, so both the development and the presentation of data models should employ them all. By the same token, a data model in a repository such as ERwin’s, can be presented in the format most suitable for a particular use. Some people might prefer a spreadsheet of data elements, someone else might prefer to see the lineage of certain data, and someone else will prefer a diagram showing just the entities and relationships.

5 – Using conceptual models as a foundation for process and requirements analysis

This is a large topic, worthy of a white paper of its own, but the basic idea is straightforward and we have touched on it already – processes and applications act on the entities in a data model. Once these entities are identified, it becomes much easier to discover what processes act on the entity, or what services and use cases manipulate the entity. For example, in a Loans Management setting at a bank, some of the most important entities were Customer, Loan, and Loan Payment. Knowing this helped the team rapidly and more completely identify the necessary processes, services, and use cases, including the following.

- Business processes – Acquire New Customer, Grant Loan, Collect Loan Payment, etc.
- Services – Register Loan Application, Open Customer Account, Issue Loan Invoice, etc.
- Use Cases – Customer Registers Loan Application, CSR Registers Loan Application, etc.

The key point is that it’s hard to describe process or functional requirements without referring to entities in a data model. Having a data model that’s easily understood and accessible makes it substantially easier as well as more complete and consistent.

Let’s now look at three examples that demonstrate how data modeling has been made useful to business audiences, including business professionals, business analysts, and business process specialists.

Case Study 1 – Data Models Help Regulatory Agency Launch New Program and Processes

A regulatory agency responsible for the safety of industrial and technical installations (e.g., industrial/commercial heating plants and oilfield operations) was implementing a new program that would allow participating client organizations to...
managing their own safety programs. In effect, they would become “self-inspecting.” The agency’s safety inspectors would place more emphasis on assuring that a program was in place and being followed, and less on inspecting the equipment and its installation.

As is often the case with new initiatives, in both public and private sectors, there wasn’t clarity on some of the fundamental policies and rules. In fact, it emerged that it wasn’t just answers that were lacking, but that some of the questions had not yet been determined. That meant it wasn’t clear how the new business processes should work, nor was it clear what changes would be needed to the supporting applications, or even if they could be changed to accommodate the new program. Data modeling, although we didn’t initially call it that (a recurring theme) was central to addressing these issues.

Our first goal was to clarify vocabulary, by discussing the business, with no reference to data or data modeling. Of course, we wanted to learn about the business, but this was also a way to determine what the most important “things” were for the new program. And, of course, these “things” would become the entities in a conceptual model. We started with a series of one-hour telephone interviews with key players from both head office and field functions. Some of our questions were:

- Could you please describe the essence of the program from your area’s perspective?
- What do you see as the benefits to the agency and to the clients, and how will they be measured?
- How will your group participate in the new program, and what will be different from current operations?
- What new information will be required to operate and manage the program? (This was as close to a “data” question as we got.)
- Are there any specific issues or obstacles you foresee?

Of course we captured the answers, but we also noted every term (noun) they used, whether it was likely a thing, a fact about a thing, a metric, a system, an organization or job title, or anything else. From experience and context, it was relatively easy to identify the core terms (“entities”) and see that there were many synonyms. For instance, terms such as “site,” “location,” “operation,” and “facility” were used to describe one of the client’s ongoing operations. Clients were variously referred to as “client,” “customer,” “operator,” “applicant,” and so on.

Once we had agreement on the fundamental “things” of the organization, we moved on to looking at the relationships among these things by developing an entity-relationship diagram, but with a more subtle approach. This took the form of business policy questions such as “does a Program apply to a Client, and therefore all of their Facilities, or to a specific Facility, or to a subset of a Facility?” It emerged that questions such as this hadn’t been considered, so we were actually helping the organization clarify the program’s policies. Data modeling helped the agency formulate the fundamentals of this important new program and introduce it without costly or embarrassing delays and missteps.

**Case Study 2 – Global Technology Company Improves Marketing & Sales Processes with Data Models**

This second example illustrates a phenomenon that my colleagues and I are encountering increasingly often - a client retains us to help with their business processes but it turns out that the real problem is in the data. That was the case at one of the world’s best-known technology companies where they were facing extremely serious problems in their marketing and sales processes. They were sure that what they needed was redesigned processes and improved application functionality, but we quickly determined that the fundamental problem was data-related, not process-related. Much of the work in the processes involved reconciling some 300 different and inconsistent data sources. You could argue that this is, in fact, a process problem - they had no data management processes! Fairly quickly, we had agreement that their business processes were
never going to work properly until they had, at a minimum, a foundation of consistent terms and definitions upon which was built an overarching, conceptual data model, which in turn would guide future development of processes and applications. In three, half-day workshops, we developed and validated the required data model, and used that to put in place a transitional development plan.

In the previous example, we worked primarily with representatives from the business, but in this case we worked with a team of experienced, technically-oriented business analysts and project leaders. They knew the systems inside out, but it soon emerged that this strength was also a weakness - they were almost unable to describe the business except in terms of how each of the company’s systems saw the business. With that many inconsistent systems, it is no wonder there was no consistent view of the business.

As usual, the first step was to establish a fundamental vocabulary (the entities,) so we brainstormed with the group for terms, and backed that up with terms culled from background documents. The ubiquitous Post-it was the primary tool. Getting ideas from the group was not a problem - the challenge was getting all of the technical references out. We pushed and pushed until we had a set of pure, business terms - Campaign, Order, Client, Account, Team, External Entity (Person or Organization,) Role, Activity, Supplier, and many others.

That was enough of a challenge, but things got really interesting when we started formulating definitions. In this case, we had to iterate between defining the terms and drawing a data model. Three key findings illustrate why:

- “Customer” turned out to not be a distinct thing, managed by the business, even though everyone talked about “the Customer.” Rather, it was a view based on selecting certain “External Entities” and “Accounts”.
- “Prospect” was spoken of as a distinct thing, but it turned out to be a status of an Account.
- The “Team” working for a Customer, on a Campaign, or on an Opportunity also turned out not to be a distinct thing - it was a set of Persons, each playing a Role with respect to some other entity such as an Opportunity, Order, External Entity, Account, or Lead.

These and other discoveries brought the central problem into sharper focus - there was no distinct thing like a Customer or a Team working its way through the business processes. Rather, the “token” in a process such as “Assess Team Performance” was essentially a report, the definition of which varied widely. Only through focusing on the conceptual data model was this realized.

Two comments by the senior manager in charge highlighted the value of the approach:

- “You accomplished more this week than we had in the previous 8 months.”
- “This is the first time this team has described the business we support without reference to systems.”

Case Study 3 – HR Organization Determines Future of Business Software through Data Modeling

A global company spent ten times an application’s purchase price on customization, but was still very unsatisfied with the support it provided to their Recruiting and Hiring processes, as were the federal regulators they were required to report to. We were brought in to advise the client on whether to stay with their (unsatisfactory) implementation, or move to the recently-released version which appeared to have improved functionality as well as (of course) a much flashier user interface.
'The dilemma, as it always is when a purchased application is heavily modified, is that migrating to the new release would mean losing millions of dollars worth of customization. The client assumed that we would assess application functionality in the form of a “gap-fit” analysis but this was another case where the problem lay elsewhere—in the data structure. How we came to that conclusion is an instructive example.

We began by reviewing problem logs and conducting interviews to discover specific complaints that HR professionals had with the application. The problems were then classified according to a three-tier systems model in which we identified whether each problem was primarily due to user interface or reporting (presentation layer), business logic (application layer), or data (data management layer). We also ranked each problem as severe, moderate, or minimal. It was immediately clear that most of the severe problems, whichever layer they manifested in, could be traced back to a data issue. We began to suspect that the real problem was that the application's underlying data model was a poor match with the business' underlying data model. Neither had been visualized, so that was next.

We were able to construct a credible conceptual data model for the business from the problem statements. For instance, if it's a problem that “the system allows multiple Hires against a single Requisition” we can conclude that the correct model is that there can only be one Hire per Requisition. Or, the complaint that “the system fails to adequately distinguish Applicants and Candidates” also tells us something important.

The next step was to confirm our impressions of how the application saw the business, so we contacted the vendor to request a data model for the application. They weren’t even sure what a data model was, but the next day a thick pile of SQL DDL arrived by courier. If only they’d sent us an ERwin model! It took a couple of days of work, much scissors and glue, and many foreign key references (thank goodness they didn’t have application-maintained relationships), but we produced a graphic of the physical data model. We then removed the obvious physical design accommodations, producing our best guess at the logical model, and then abstracted this up to a conceptual model. Reflecting the 1:5 guideline, this reduced a 100 entity logical model to a roughly 20 entity conceptual model.

Looking at the business’ conceptual model and the application’s conceptual model side-by-side, the differences were remarkable. This difference was the best way to illustrate the essence of the situation, so we scheduled a presentation of our findings for the core team and senior management. How we conducted that data model review was key to the success of this assignment.

Rather than use the term “data model” we opened the presentation by saying “here’s how we think you see the world.” In fact, we called it a “world view,” as we often do. At one of the two large whiteboards, we drew the boxes and lines of the data model accompanied by a narrative that was pure business - “We work in a global organization characterized by complex organizational structures built from legal entities, SBUs (Strategic Business Units), Geographies, Divisions, and so on. Within each Organizational Unit are Positions, each of which may have an Employee assigned to it, or be vacant.” After 10 minutes or so, one participant pronounced this “the clearest description of our business I’ve ever seen” and asked to have our materials for use in the new employee onboarding process.

A video recording of this session would have been priceless. As we unveiled the “world view” of the core application, jaws progressively dropped. The difference between how the application saw the world and how the business saw the world was huge. The highlight was when the senior executive attending said, “This has been a revelation.” After unsuccessful negotiations with the vendor, the ultimate decision was to abandon the application entirely. That doesn’t sound like a classic definition of “success,” but it avoided years of continued frustration and unnecessary expense.
Conclusion: Common Success Factors

These examples, just a few drawn from 30 years of business-oriented data modeling, demonstrate that simple techniques go a long way to getting the business involvement that is so essential to effective data modeling. Two factors in particular stand out in these three cases. First, the focus was on the business goals, and the project team members were not aware that data modeling was a technique that was going to help them with their problem. They either weren’t familiar with data modeling, or thought it was a technical discipline not useful for business analysis. Second, we didn’t try to convince anyone that data modeling was “the answer” – we “just did it,” almost in a stealth fashion. Only after successfully using data modeling did we begin to introduce the language and constructs of data modeling, to people who were already receptive because a real problem had been solved.

To recap, some of the core techniques we have discussed include:

1. Use the language of the business, not technical jargon. Remember, you’re describing a business, not a database.
2. Use the right sort of model for the audience, i.e. a business-friendly conceptual model that highlights essential business concepts for business users, and a detailed logical model with keys, data types, nullability, etc. for developers and DBAs. Always think about whom you are speaking to.
3. Make information easily accessible to the audience. Publish information on the web, for example, and consider using non-modeling formats to get your audience’s attention (spreadsheet-style interface, keyword searching, etc.)
4. Follow graphic principles that will make your data model diagrams easier for all audiences to understand. Make it even better with scenarios, sample data, schematic diagrams, and other aids.
5. Make sure your data modeling approach “plays nicely with others.” The key is to integrate your techniques with those used by, for example, Business Analysis and Business Process Improvement professionals. Strive to share the information in your models so that they are used across the enterprise. Store your models in a common repository so that all your hard work can be shared and reused across teams and projects. Share and share alike, to truly get the value out of your efforts, and make data models a “business as usual” tool.

Keep these points in mind, and remember the lessons from the case studies, and you’ll be well on your way to “getting down to business” with data modeling.