

Business analysis. introduction to data modeling

[Business](#)



Before you sit down in front of the keyboard and start creating a database application, it is critical that you take a step back and consider your business problem-? in this case, the kitchen supply scenario presented in Lesson 2-? from a conceptual point of view. To facilitate this process, a number of conceptual modeling techniques have been developed by computer scientists, psychologists, and consultants. For our purposes, we can think of a conceptual model as a picture of the Information system we are going to build. To use an analogy, conceptual models are to Information systems what blueprints are to buildings.

There are many different conceptual modeling techniques used in practice. Each technique uses a different set of symbols and may focus on a different part of the problem (e. G. , data, processes, information flows, objects, and so on). Despite differences in notation and focus, however, the underlying rationale for conceptual modeling techniques is always the Michael Bryon (Ca) Last update: 02-May-01 1 of 23 An Introduction to data modeling Introduction: The importance of conceptual 3. 1. 1. 1 Entities and attributes What is data modeling?

A data model is a simply a diagram that describes the most important " things" in your businessenvironmentfrom a data-centric point of view. To illustrate, consider the simple RED shown in Figure 3. 1 . The purpose of the diagram is to describe the relationship between the data stored about products and the data stored about the organizations that supply the products. FIGURE 3. 1: An RED showing a relationship between products and suppliers. The rectangles in Figure 3. 1 are called entity types (typically shortened to " entities") and the ovals are called attributes.

<https://assignbuster.com/business-analysis-introduction-to-data-modeling/>

The entities are the " things" in the business environment about which we want to store data. The attributes provide us with a means of organizing and structuring the data. For example, we need to store certain information about the products that we sell, such as the typical selling price of the product (" Unit price") and the quantity of the product currently in inventory (" Sty on hand"). These pieces of data are attributes of the Product entity. It is important to note that the precise manner in which data are used and processed within a particular business application is a separate issue from data modeling.

For example, the data model says nothing about how the value of " Sty on hand" is changed over time. The focus in data modeling is on capturing data about the environment. You will learn how to change this data (e. G. Process orders so that the inventory values are updated) once you have mastered the art of database design. Product Unit price Sty on hand Product Entity supplied by Cardinality Attributes Supplier Relationship Name Address A data modeled assumes that if the right data is available, the other elements of the application will fall into place effortlessly and wonderfully. For now, this is a good working assumption.

Introduction: The importance of conceptual environment in which your wholesale company operates. However, it is easy to imagine a different environment in which each product is supplied by multiple suppliers. For example, many suppliers may carry a particular brand of wire whisk. When you run out of whisks, it is up to you to decide where to place your order. In other words, it is possible that a many-to-many relationship exists between

suppliers and products. If multiple supplier exist, attributes of the product, such as its price and product number may vary from supplier to supplier.

In this situation, the data requirements of a many-to-many environment are slightly more complex than those of the one-to-many environment. If you design and implement your database around the one-to-many assumption but then discover that certain goods are supplied by multiple suppliers, much effort is going to be required to fix the problem. In addition to entities and attributes, Figure 3. 1 shows a relationship between the two entities using a line and a diamond. The relationship construct is used-? not surprisingly-? to indicate the existence or absence of a relationship between entities.

A crow's foot at either end of a relationship line is used to denote the cardinality of the relationship. For example, the crow's foot on the product side of the relationship in Figure 3. 1 indicates that a particular supplier may provide your company with overall different products, such as bowls, spatulas, wire whisks and so on. The absence of a crow's foot on the supplier side indicates that each product in your inventory is provided by a single supplier. Thus, the relationship in Figure 3. 1 indicates that you always buy all your wire whisks from the same company. . 1. 1. 3 Modeling assumptions

The relationship shown in Figure 3. 1 is called one-to-many: each supplier supplies many products (where many means " any number including zero") but each product is supplied by one supplier (where " one" means " at most one"). The decision to use a one-to-many relationship reflects an assumption about the business. Herein lies the point of drawing an ERD: The diagram makes your assumptions about the relationships within a particular business environment explicit before you start building things. The role of the

<https://assignbuster.com/business-analysis-introduction-to-data-modeling/>

modeled 3. 1. 1. In the environment used in these tutorials, you are the user, the designer, and the implementer An introduction to data modeling of the system. In a more realistic environment, however, these roles are played by different individuals (or groups) with different (programmers, database specialists, and so on) is that they seldom leave their busbies to communicate with end-users of the software they are writing. Similarly, it is generally safe to assume that users have no interest in, or understanding of, low- level technical details (such as the cardinality of relationships on Reds, mechanisms to enforce referential integrity, and so on).

Thus, it is up to the business analyst to bridge the communication gap between the different groups involved in the construction, use, and administration of an information system. As a business analyst (or more generally, a designer), it is critical that you walk through your conceptual oodles with users and make sure that your modeling assumptions are appropriate. In some cases, you may have to examine sample data from the existing computer- based or manual system to determine whether (for instance) there are any products that are supplied by multiple suppliers.

At the modeling stage, making changes such as converting a one-to-many relationship to a many-to-many relationship is trivial-? all that is required is the addition of a crows foot to one Introduction: The importance of conceptual end of the relationship, as shown in Figure 3. 2. In contrast, making the same change once you have implemented tables, lilt a user interface, and written code is a time-consuming and frustrating chore.

FIGURE 3. 2: An RED for an environment in which there is a many-to-many relationship between products and suppliers.

<https://assignbuster.com/business-analysis-introduction-to-data-modeling/>

Product Unit price Sty on hand The addition of a second cross foot transforms the one-to-many relationship into a many-to-many relationship. Supplier Generally, you can count on the 10x rule of thumb when building software: the cost of making a change increases by an order of magnitude for each stage of the systems development lifecycle that you complete. An introduction to data modeling Introduction: The importance of conceptual 3. 1. 2. 1 Entities 3. 1 . 2 Data modelers typically adopt a set of notational conventions so that their diagrams are consistent.

For example, large IT organizations and consultancies typically adopt a methodology - a set of tools and procedures for applying the tools that specifies the notation used within the organization. Enforcing standardization in this way facilitates teamwork on large projects. Similarly, if a computerized software engineering (CASE) tool is used for conceptual modeling and design, notational conventions are often enforced by the software. What follows is a brief summary of the notational conventions that I use when drawing ERDs.

Keep in mind, however, that ERDs are first and foremost a tool for communication between humans. As such, the precise notation you use is not particularly important as long as people can read and understand the diagrams. With experience, you will come to realize that differences in the shapes of the boxes and lines have little effect on the core concepts of data modeling. Entities are drawn as rectangular boxes containing a noun in singular form, as shown in Figure 3. 3. FIGURE 3. 3: An entity named "Customer". Customer

You will see later that each entity you draw ultimately becomes a table in your database. You might want to keep this transformation from entity to table in mind when selecting the names of your entities. For example, your entity names should be short but descriptive.

3. 1. 2. 2 Relationships

A relationship between entities is drawn as a line bisected by a diamond. The diamond contains a verb (or short verb phrase) that describes the nature of the relationship between the entities, as shown in Figure 3. 4. Named relationships are used to make the ERDs more readable.

However, unlike entity names, relationship names never show up in the final database. Consequently, it does not really matter how you label your relationships, as long as it can be argued that the term "method" is grammatically preferable. In Europe, for example, the term "method" tends to be favored.

Introduction: The importance of conceptual modeling

Generally, ERDs make certain assumptions about the reader's knowledge of the underlying business domain.

FIGURE 3. 4: A relationship named "buys". As the labels make the diagram easier to interpret.

To illustrate, consider the relationship between products and suppliers shown in Figure 3. 1. The relationship is described by the verb phrase "supplied by". Although one could have opted for the shorter relationship name "has" instead, the resulting diagram (e. g. , "Supplier has product") would be more difficult for readers of the diagram to interpret.

3. 1. 2. 3 Relationship direction

A notational convention supported by some CASE tools is to require two names for each relationship: one that makes sense in one direction (e. g. , "is supplied by"), and another that makes sense in the opposite direction (e. g. , "supplies"). Although double-naming may make the diagram easier

<https://assignbuster.com/business-analysis-introduction-to-data-modeling/>

to read, it also adds clutter (twice as many labels) and imposes an additional burden on the modeled. Cardinality 3. 1 . 2. 4 One issue that sometimes troubles neophyte data meddlers is that the direction of the relationship is not made explicit on the diagram. Returning to Figure 3. 1, it is obvious to me (since I drew the diagram) that the relationship should be read: " Product is supplied by supplier. Reading the relationship in the other direction (" Supplier is supplied by product") makes very little sense to anyone who is familiar with the particular problem domain. As discussed in Section 3. 1. 1. 2, the cardinality of a relationship constrains the number of instances of one entity type that can be associated with a single instance of the other entity type. The cardinality of relationships has an important impact on number and structure of the tables in the database. Consequently, it is important to get the cardinality right on paper before starting the implementation.

An introduction to data modeling There are three fundamental types of cardinality in Red's: ; One-to-many -? You have already seen an example of a one-to-many relationship in Figure 3. 1 . You will soon discover that one-to-many relationships are the bread and butter of relational databases. One-to-one -? At this point in your data modeling career, you should avoid one-to-one relationships. To illustrate the basic issue, consider the RED shown in Figure 3. 5. Based on an existing paper-based system, the modeled has assumed that each customer is associated with one " customer record" (i. . , a paper form containing information about the customer, such as address, fax number, and so on). Clearly, each customer has only one we automate the system and get rid of the paper form, then there is no reason not to combine the Customer and Customer Record entities into a single entity called

Customer. Introduction: The importance of conceptual FIGURE 3. 5: An incorrect one-to-one relationship associated with Customer Record In many cases, one-to-one relationships indicate a modeling error. When you have a one-to-one relationship such as the one shown in Figure 3. , you should combine the two entities into a single entity. ; Many-to-many -? The world is full of monotony-many relationships. A well-used example is " Student takes course. " Many-to-many relationships also arise when you consider the history of an entity. To illustrate, consider the RED shown in Figure 3. 6. At first glance, the relationship between Family and Single-Family Dwelling (SF) might seem to be one-to-one since a particular family can only live in one SF at a time and each SF can (by definition) only contain a single family. However, it is possible for a family to live in different houses over time.