Home (/) / SQL (/SQL/) / Database design for mere mortals (/SQL/Database+design+for+mere+mortals/)

109

What Are Business Rules?

◀ Previous Next ▶

A business rule is statement that imposes some form of constraint on a specific aspect of the database, such as the elements within a field specification for a particular field or the characteristics of a given relationship. You base a business rule on the way the organization perceives and uses its data, which you determine from the manner in which the organization functions or conducts its business.

An important aspect of any design process is making choices. In database design, for example, you must choose which data to store in the database; you would not necessarily want or need to store every last piece of data the organization might possibly use. The data you finally choose to store and how you decide to store it will be determined by the way the organization uses its data. A hospital may wish to store times of various events to the second, whereas a warehouse requires only the date for any given event.

To guide these and other choices you'll be required to make during the database-design process (and later, when you implement the database in an RDBMS), you need a formal statement of the organization's business rules. These rules will influence a wide variety of database issues, such as the data you collect and store, the manner in which you define and establish relationships, the types of information that the database can provide, and the very security and confidentiality of the data itself. It is next to impossible to create a generic set of business rules that could apply to two or more organizations. Each organization has its own data and information requirements, and each has its own unique way of conducting its business; therefore, every organization needs its own specific set of business rules.

The following statement is an example of a typical business rule:

> A SHIP DATE cannot be prior to an ORDER DATE for any given order.

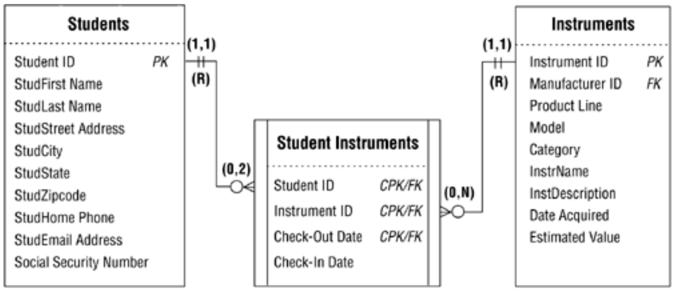
This particular business rule imposes a constraint on the Range of Values element of the field specifications for a SHIP DATE field. It will help ensure that the value of SHIP DATE is meaningful within the context of a sales order. Without this constraint, you could enter any date into the field (including one prior to the ORDER DATE), making the SHIP DATE field's value absolutely meaningless. The business rule is what makes the SHIP DATE field's value contextually meaningful.

Because business rules depend on the manner in which an organization perceives and uses its data, it is quite possible that a particular rule can be used by several organizations, but for completely different reasons.

For example, say that the music department at Bel Air High School is known far and wide for the quality of musicianship it develops in its student musicians. The students are able to attain this level of musicianship because they're encouraged to focus their musical studies and restrict themselves to learning no more that two instruments. In another part of town, the music department at Lake City High School (a private school) also imbues its student musicians with a high quality of musicianship by helping the students focus their musical studies. The students at this school, however, are restricted to learning no more than two instruments due to school policy; the school's inventory of musical instruments is very limited.

Coincidentally, both schools are in the process of designing their own database. In each case, the school will use the database to support its daily operations and administrative functions. It so happens that each database contains the tables shown in Figure 11.1.

Figure 11.1. Tables from the Bel Air High School and Lake City High School databases.



Both schools are at the same stage of the database-design process and are currently establishing business rules. As it turns out, each school is using the following business rule in their respective databases:

> A student cannot have more than two instruments checked out at the same time.

This business rule applies to the degree of participation between the STUDENTS table and STUDENT INSTRUMENTS table. In this instance, a single record in the STUDENTS table cannot be associated with more than two records in the STUDENT INSTRUMENTS table where the value of CHECK-IN DATE for each record is null; a null value in the CHECK-IN DATE field indicates that the instrument is still in the student's possession.

The rule does apply to both schools, yet each school requires it for a different reason. Bel Air High School requires the rule because of the manner in which its music program has been established, whereas Lake City High School requires the constraint because of the physical limitations of its instrument inventory. The fact that both schools developed an identical rule is pure coincidence. This example illustrates both that a business rule is, indeed, based on the way an organization functions or conducts its business and why every organization must have its own specific set of business rules.

The example also illustrates another issue: You cannot establish constraints imposed by certain business rules, such as this one, within the logical design of the database. For instance, there is no clear way for you to indicate that the CHECK-IN DATE values must be tested in order to determine whether a student can check out another instrument. You must instead address and establish the constraint outside of the logical design of the database. How do you determine whether you can properly represent a given constraint within this process? You do so by identifying the type of business rule you're defining.

Types of Business Rules

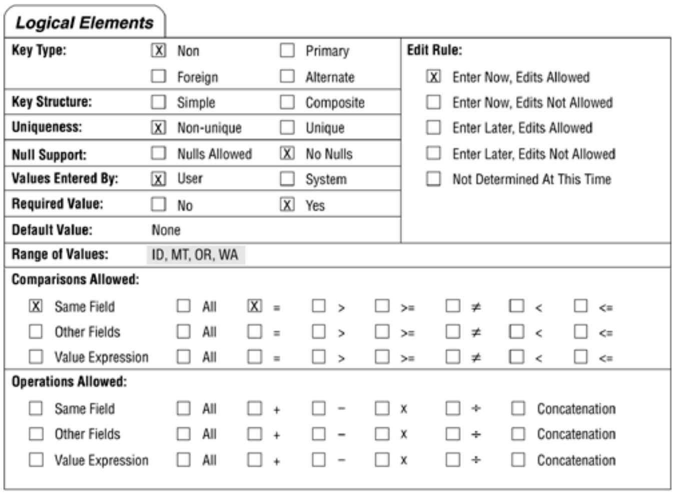
There are two major types of business rules: database oriented and application oriented. Both types of business rules impose some form of constraint and help enforce and maintain overall data integrity, but they differ with regard to where and how they are established.

Database oriented business rules impose constraints that you can establish within the logical design of the database. You implement a given constraint by modifying various field specification elements, relationship characteristics, or a combination of the two. The statement from which you derive the constraint is a database oriented business rule if you can meaningfully and clearly establish the constraint by either of these means. For example, say you have a VENDORS table and define the following business rule for the VENDSTATE field in that table:

> We conduct business exclusively with vendors from the Pacific Northwest.

This business rule limits the values that you can enter into the VENDSTATE field to WA, OR, ID, and MT. You can establish the business rule's constraint in a meaningful manner by modifying the Range of Values element in the field specifications for the VENDSTATE field. Figure 11.2 shows the modification.

Figure 11.2. Implementing a constraint imposed by a database oriented business rule.



Logical Elements

Key Type:	☒ Non ☐ Foreign	☐ Primary ☐ Alternate	**Edit Rule:** ☒ Enter Now, Edits Allowed ☐ Enter Now, Edits Not Allowed ☐ Enter Later, Edits Allowed ☐ Enter Later, Edits Not Allowed ☐ Not Determined At This Time
Key Structure:	☐ Simple	☐ Composite	
Uniqueness:	☒ Non-unique	☐ Unique	
Null Support:	☐ Nulls Allowed	☒ No Nulls	
Values Entered By:	☒ User	☐ System	
Required Value:	☐ No	☒ Yes	
Default Value:	None		
Range of Values:	ID, MT, OR, WA		

Comparisons Allowed:

☒ Same Field	☐ All	☒ =	☐ >	☐ >=	☐ ≠	☐ <	☐ <=
☐ Other Fields	☐ All	☐ =	☐ >	☐ >=	☐ ≠	☐ <	☐ <=
☐ Value Expression	☐ All	☐ =	☐ >	☐ >=	☐ ≠	☐ <	☐ <=

Operations Allowed:

☐ Same Field	☐ All	☐ +	☐ −	☐ x	☐ ÷	☐ Concatenation
☐ Other Fields	☐ All	☐ +	☐ −	☐ x	☐ ÷	☐ Concatenation
☐ Value Expression	☐ All	☐ +	☐ −	☐ x	☐ ÷	☐ Concatenation

Application oriented business rules impose constraints that you cannot establish within the logical design of the database. You must instead establish them within the physical design of the database or within the design of a database application, where they will be more applicable and meaningful. (I use the term database application here to refer to a program written in some RDBMS software that allows people in the organization to use the database easily and to perform tasks related to their daily work activities.)

Here is an example of a typical application oriented business rule:

> A customer with a "Preferred" status receives a 15% discount on all purchases.

This business rule determines the amount of discount applied to a customer's purchases, based on a particular status. You cannot establish this constraint meaningfully in the logical design for two reasons: There is no field in which to store the discount amount (the amount is a result of a calculation, and calculated fields are not allowed in a table), and there is no way to indicate the criterion usedthe customer's statusto determine the discount. This is a rule that you must establish within the physical design of the database or the design of the database application.

Note

The manner in which you actually define and establish application oriented business rules is a topic that is beyond the scope of this book. Some RDBMSs provide tools that allow you to implement common application oriented business rules relatively easily; most RDBMSs will require you to write programming code to implement and enforce these rules.

Although both types of business rules are important, you'll focus on database oriented business rules during this stage of the database-design process.

Note

Throughout the remainder of the book, I'll refer to database oriented business rules simply as business rules.

0 Comments etutorials

1 Login

♡ Favorite 8 Tweet Share Sort by Best

Start the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

Be the first to comment.