# Data Types and Other Issues

Most databases allow a variety of types. Typically, fields can be designated as string, numeric, data/time, and autonumbering for key fields. Sometimes there is a choice between various types of numbers.

**Numeric:** This should be used for data that will be used in numeric calculations only. Fields like Social Security number, phone number and Zip code contain digits, but they are not numeric. You aren't going to add to the social security number or find the average of phone numbers. They should be designated as string data. Two additional questions that indicate a field is string rather than numeric: leading zeros are important, and the data is often formatted in a particular way, such as Social Security numbers that are formatted as "123-45-6789" Numbers can be integer (whole numbers) or decimal numbers, also called float or long. Integers are things that are usually counted, while decimals include things that can be measured and any monetary values.

**String**: This should be used for text, including data that has digits but is not used for numeric operations. Digits that are always formatted the same way, such as Social Security numbers that are formatted as "123-45-6789" should be stored without the formatting: "12345789"

**Names**: In English we refer to first and last names. In many cultures, these are referred to as given name and family name. Asking someone their last name can cause confusion. Names can be stored as family name, given name or as two separate fields. If separate fields are used, there must also be a field for a suffix (items that appear after the name such as 3rd, Jr. etc.) Storing a name as "first last" is an error as it does not allow the names to be sorted in the standard alphabetical order.

**Age**: Instead of storing age, store the year born. If you store the age as 12, it will still be 12 five years later. In general, if a value can be calculated, store the value it is calculated from, not the current value.

**Addresses**: As our world becomes more global, consider that you may need to include country as part of the address. In addition, not all countries use a numeric postal code such as the American zip code. All parts of an address should be designated string fields.

**Boolean**: Boolean values are ones that are true or false. Before designating a field as Boolean, consider whether there are additional choices. For instance married, might be better as a string, allowing the user to select single, married, divorced, widowed, etc. If you decide that a field is Boolean, and Boolean is not one of the types available, it is usually designated as integer, with 0 representing false, and any other value as true, but usually 1 is used for true.

**Date/Time**: A date/time field usually has methods to extract each part of the field. Often you know part of a date. However, most databases won't allow an invalid date and you must fill out all parts. For instance you may know the year, but not the month and day, or vice-versa. One way of handling this is to use January 1 when only the year is known. Use 1900 if the year is not know. The thinking is that no one born in 1900 is still living and no business is conducted on January 1. If necessary, add another field to tell if the date is exact or not. You can also use three separate integers for month, day, and year.