**Data Validation in Access**

An organization keeps its data on the computer in the form of a database. This data is converted to information by extracting items that a specific individual needs to do his or her job. Different people need different things extracted from the huge database; information is different things to different people. It is important that when the data is entered into the database that it does not contain errors. It is not that all errors can be prevented from entering the database, but an attempt should be made to capture as many as possible. The errors will eventually have to be corrected. For example, if the grade assigned to a student is an invalid grade (example ‘Q’), it should be and can be caught immediately. If a student who earns an A is assigned an F, the student will question it and it will need to be changed. The purpose of storing data in a database is to provide information to various people when, how, and where they need that information. The quality of the output depends on the quality of the input. There is an expression for this: Garbage In, Garbage Out (GIGO). If someone finds too many mistakes in the output from a database, he or she will stop trusting it, and might even stop using it.

To prevent bad data from being entered into the database table, there are certain standard validation processes that can be performed on an item being entered. Access has some of these features built into it. Some of them are listed below:

1. Range check: certain values have to be within a certain range. Examples:
   1. A student’s GPA has to be between 0 and 4.0.
   2. The number of hours worked by an employee in a week is normally 40, maybe 60 maximum with overtime, but definitely cannot be more than 168 (the number of hours in one week).
2. Value check: data in certain fields has to contain certain specific values. Here are some examples:
   1. Grades: valid grades are A, B, C, D, F, I, W, and maybe CR or NC. A grade of Q is not acceptable.
   2. The fifty states of the United States have standard codes used by the Post Office. CA for California, CL is not acceptable.
   3. The list of majors within a college: Engl, CIS, Phys, Chem, Math, etc. PQRS is not acceptable. If the list is short, it can be itemized within the table design. If the list is very long, there are many ways to handle it including creating a table for the list.
3. Dates: Access has a field type called Date. It will not allow you to enter an incorrect date for a specific month, for example: Nov. 31 or Feb 29 for a non-leap year. We also want to make sure that certain dates are after certain other dates. For example: if an item was charged on a certain date, the due date of the amount cannot be before the charge date. Date calculations can also be performed in Access: if the current date is more than 30 days (say) after the charge date, a reminder can be sent to the customer to pay the bill.
4. Class test: whether a certain field contains the correct type of information, numeric or alphabetic. Names should generally not contain numbers, and numeric fields like telephone numbers should not have characters. But there are fields like the license plate number which contains both, but contain certain patterns of numbers and characters which can also be checked.
5. Consistency check between two related fields:
   1. A return date cannot be before a departure date
   2. A male gender cannot have a pregnancy code of yes
6. Existence check: all data that needs to be present should be present, no blanks should be acceptable. Certain values can be Null, for example a new student who has not yet been assigned an advisor would have that field Null for the time being. For Sex: should be either M or F, a blank should not be used to indicate one or the other.

Access has many of these features available under the design view of the table. Take a look at the sample database I have provided, RCCSIS.accdb. It has two tables, Students and Advisors. The major code and the GPA value have validation procedures set. There is a 1: M relationship between advisors and students (one advisor advises many students). The PK of the Advisor is stored as a Foreign Key in the (many) Student table to set up a relationship. It is critical that the PK and the FK have identical values for the relationship to work. Referential Integrity will make sure that students are not assigned a nonexistent advisor. There is another note I have placed on the home page titled Relationships – Required Reading. Please read that.

The reference tab allows the database designer to point to another table of values if the list is very long, for example the codes for the fifty states of the US. This helps in the validation of a list of codes, and that bad data like CL for California does not creep into the database. Combo boxes can display the list so that the person entering the data does not have to type in the data but needs to select the value.

A final note: computers do not automatically generate error messages. A human programmer or software specialist writes those error messages. Error messages should have two features: they should be user friendly, and they should be informative in how to fix the error and proceed. An error message like “you made a mistake” is neither. Access has this feature available under the design view of the table. Look at the examples in the RCCSIS database sample.