

VALIDATION RULE EXAMPLES

The following table provides examples of field-level and record-level validation rules, plus explanatory validation text. You can adapt these examples to fit your content.

| VALIDATION RULE | VALIDATION TEXT |
|----------------------------------|-----------------------------------------------------------------------------------------------------|
| <>0 | Enter a nonzero value. |
| >=0 | Value must be zero or greater. -or- You must enter a positive number. |
| 0 or >100 | Value must be either 0 or greater than 100. |
| BETWEEN 0 AND 1 | Enter a value with a percent sign. (For use with a field that stores number values as percentages). |
| <#01/01/2007# | Enter a date before 2007. |
| >=#01/01/2007# AND <#01/01/2008# | Date must occur in 2007. |
| <Date() | Birth date cannot be in the future. |
| >=Int(Now()) | Enter today's date. |
| M Or F | Enter M for male or F for female. |

Validation Rules for fields

This rule is applied when you enter data into the field. You cannot tab to the next field until you enter something that satisfies the rule, or undo your entry.

Examples:

| To do this ... | Validation Rule for Fields | Explanation |
|--------------------------------|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| Accept letters (a - z) only | Is Null OR Not Like "[!a-z]*" | Any character outside the range A to Z is rejected. (Case insensitive.) |
| Accept digits (0 - 9) only | Is Null OR Not Like "[!0-9]*" | Any character outside the range 0 to 9 is rejected. (Decimal point and negative sign rejected.) |
| Exactly 8 characters | Is Null OR Like "????????" | The question mark stands for one character. |
| Exactly 4 digits | Is Null OR Between 1000 And 9999 | For Number fields. |
| | Is Null OR Like "####" | For Text fields. |
| Positive numbers only | Is Null OR >= 0 | Remove the "=" if zero is not allowed either. |
| No more than 100% | Is Null OR Between -1 And 1 | 100% is 1. Use 0 instead of -1 if negative percentages are not allowed. |
| Not a future date | Is Null OR <= Date() | |
| You must fill in <i>Field1</i> | Not Null | Same as setting the field's <i>Required</i> property, but lets you create a custom message (in the <i>Validation Text</i> property.) |

| | | |
|---------------------------|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Limit to specific choices | Is Null OR "M" Or "F" | It is better to use a lookup table for the list, but this may be useful for simple choices such as Male/Female. |
| | Is Null OR IN (1, 2, 4, 8) | The IN operator may be simpler than several ORs. |
| Yes/No/Null field | Is Null OR 0 or -1 | The Yes/No field in Access does not support Null as other databases do. To simulate a real Yes/No/Null data type, use a Number field (size Integer) with this rule. (Access uses 0 for False, and -1 for True.) |

Validation Rules for tables

In addition to those rules, the following table shows the common arithmetic operators and provides examples of how you can use them.

| OPERATOR | FUNCTION | EXAMPLE |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NOT | Tests for converse values. Use before any comparison operator except IS NOT NULL. | NOT > 10 (the same as <=10). |
| IN | Tests for values equal to existing members in a list. Comparison value must be a comma-separated list enclosed in parentheses. | IN ("Tokyo","Paris","Moscow") |
| BETWEEN | Tests for a range of values. You must use two comparison values — low and high — and you must separate those values with the AND separator. | BETWEEN 100 AND 1000 (the same as >=100 AND <=1000) |
| LIKE | Matches pattern strings in Text and Memo fields. | LIKE "Geo*" |
| IS NOT NULL | Forces users to enter values in the field. This is the same as setting the Required field property to Yes . However, when you enable the Required property and a user fails to enter a value, Access displays a somewhat unfriendly error message. Typically, your database is easier to use if you use IS NOT NULL and enter a friendly message in the Validation Text property. | IS NOT NULL |
| AND | Specifies that all the data that you enter must be true or fall within limits that you specify. | >= #01/01/2007# AND <=#03/06/2008# NOTE You can also use AND to combine validation rules. For example: NOT "UK" AND LIKE "U*" . |
| OR | Specifies that one or more pieces of data can be true. | January OR February |
| < | Less than. | |
| <= | Less than or equal to. | |
| > | Greater than. | |
| >= | Greater than or equal to. | |
| = | Equal to. | |
| > | Not equal to. | |