

Setting “Custom Conditions” in Student Manager

Tips and examples:

By Chuck Havlicek

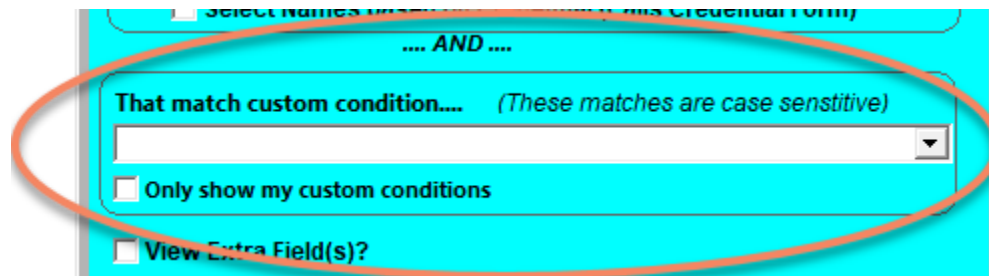
There are several places where a user can set a Custom Condition for display or report of records: The F2, F3, F5, F7, and F9 are just a few. You also can set custom parameters in areas like Duplicate Names Report and Mass Update/Change. The examples listed may also be used in the SM report system, when using the “Print When” option (“print only when expression is true”.)

While users may certainly contact their technician for help in creating these custom elements, they are welcome to put in their own custom criteria. This white paper will help guide a user through their Do It Yourself (DIY) process. NOTE: While in the F keys searches-there are no permanent changes made to the system (reports only.) When working in Duplicate Name report and Mass Update, those WILL modify data, and so when using those, you probably SHOULD (until you consider yourself an expert in this area) ask for guidance from your technician.

NOTE: If you are a SQL user, you should reference the SQL version of this document!

Overview:

You can think of “Custom Conditions “ as a build-on-the-fly query. What you are doing is giving Manager instructions about exactly WHAT records you want to return. While most of your normal options would be listed in the “Stock” data fields listed, you sometimes want to search for some unique field value or different combination; hence your ability to enter “Custom Condition” (hereafter referred to as CC).



(Example of the Custom Condition Box on Name Finder)

The Equation to solve:

What you enter in the CC box is an equation, that if evaluated to TRUE, will pull/report that record. So if in Names (F5) , you want Canadians, you'd type: NMCNTRY='CANADA' If true (the name had a country named: “CANADA”) the record will be returned. For a course (F2), if you wanted a certain Subject code (“ACEWARE”) you'd type in: COSUBCODE='ACEWARE'.

Basic Concepts:

Data Types: There are five data types in Student Manager files: Character, Date, Numeric, Logical, and Memo. There are different rules in how you handle each type. In general Character and Memo have similar options, Date and Number have similar options, and Logical is unique. *To determine the type of a data field, go to the Help Guide and click on the "Screen Layout" link.* Select the table you want to search and click on the field to get the field "Specs". If you've built queries, you'll be somewhat familiar with the differences in field types (when you get to the Step 2: "Pick an Operator").

To CAPITALIZE or not to capitalize:

For data field names, and for function names, case does NOT matter. So, using Nmname1 or NMNAME1 or EMPTY(NMNAME1) or empty(NMNAME1) is same. However, whenever you are entering text (for the compare value), you must enter the text you want in with the same capitalization as the text in the database. So, if you want names with first name of "Charles" you'll need to use the same format in your search: nmname1='Charles'

A quote by any other format...

For character data matches, your style of quote doesn't matter. You just have to use matching quotes: so single ', Double " or brackets [and] are all ok. E.g.: NMCITY='Reno', NMCITY="Reno" or NMCITY=[Reno] all work the same.

Main Data Types and evaluation operators

Character: Most of your SM data fields are character, so these will be your most common options:

Action	Operator	Example
Begins with or equals:	=	Nmname3='Smith'
Does NOT begin with	<>	nmname3<>'Smith'
Greater than	>	nmzip > "66502"
Less Than	<	nmzip < "66502"
Greater than or =	=>	nmzip => "66502"
Less than or =	<=	nmzip <= "66502"
Exactly Matches	==	nmname3=='Smith'
Between two values		Between(nmzip,'66502','66509')
		BETWEEN(<field>,"<lower value>","<upper value>")
Contains text	<text> \$ <Field>	"ACE" \$ NMFIRM
Does NOT Contain		
Text	* NOT <text> \$ f<field>	NOT "ACE" \$ NMFIRM
Is an empty field	Empty(<field>)	empty(NMNAME3)
Is NOT an empty		
Field	*NOT empty(<field>	NOT empty(NMNAME3) or ! empty(nmname2)

- NOTE: the exclamation point (!) can be substituted for the verb NOT (to save space)

Date: To enter a date for a match value, you must enter the date in a way SM can recognize. You do that with the CTOD() function. The format for date entry is: CTOD('MM/DD/YYYY') so if you wanted to search for a date added of July 1, 2019, you'd enter: COADDATE = CTOD('07/01/2019')

Using Greater/Less than: The examples above are numbers (in a character field) BUT you can also use Greater/less than for regular character fields.. i.e. NMNAME3... if you wanted to have a set of names from A to L... (separate out a list of names by alpha group) you could use NMNAME >="A" and NMNAME3 < "L"

A useful variable here is DATE(). This is a VFP function that stores the current system date. So if you wanted to look for course records entered today, you could just use: COADDATE=DATE(). What's nice about it is that if you wanted to look for records entered in the last 30 days, you could just enter COADDATE > DATE() -30. (report any record added within the last 30 days).

Action	Operator	Example
Date equals:	=	PYDATE = CTOD('07/01/2019')
Date Does NOT equal	<>	PYDATE <> CTOD('07/01/2019')
Greater than	>	PYDATE > CTOD('07/01/2019')
Less Than	<	PYDATE < CTOD('07/01/2019')
Between two values	Between(Pydate,ctod('07/01/2019'),ctod('07/30/2019'))	BETWEEN(<field>,"<lower value>","<upper value>")
Greater than or =	=>	PYDATE = > CTOD('07/01/2019')
Less than or =	<=	PYDATE <= CTOD('07/01/2019')
Is an empty field	Empty(<field>)	empty(PYDATE)
Is NOT an empty Field	NOT empty(<field>)	NOT empty(Pydate)

Numeric: The Numeric comparison operators are the same as the Date ones, except that in numeric operations, you can't search for an "Empty" field. (Presumably you would be looking for a value of zero or NOT zero).

Logical: Logical variables are TRUE or FALSE. This can be represented by the field itself.. or by using a comparison of =.T. or =.F. So to report All cancelled registrations you could put in:

RGCANCEL (would return ONLY ones where RGCANCEL is TRUE)
or
RGCANCEL = .T.

To report all NON cancelled reggies, you'd type

NOT rgcancel (or ! Rgcancel) (would return records are NOT Cancelled - RGCANCEL is FALSE)
or
RGCANCEL=.F.

Memo: Memo fields (the big “Blob” text fields) fewer options

Action	Operator	Example
Contains text	<text> \$ <Field>	“Contract” \$ Conotenprt
Does NOT Contain Text	NOT <text> \$ <field>	NOT ““Contract” \$ Conotenprt
Is an empty field	Empty(<field>)	empty(Conotenprt)
Is NOT an empty Field	NOT empty(<field>)	NOT empty(conotenprt)

Other Useful Functions:

UPPER() The UPPER() function is useful when searching for text when the text may be entered with different capitalization:

The UPPER() function will evaluate the target field as if it was all uppercase. So if you want to find the word “contract” in name comments.. and you are NOT sure how it was capitalized, use:

```
“CONTRACT” $ UPPER(nmcomm)
```

That will look for the letters CONTRACT.. no matter how the word might have been capitalized (NOTE: this how the SM Query system searches for character values... it ignores case).

BETWEEN() Returns records where a specific data field is between two values... can be used with character, date, and numeric fields.

Examples: BETWEEN(RGREGFEE,49,79) Returns registrations with fee between 49 and 79
 BETWEEN(RGADDDATE,CTOD('01/01/2019'),CTOD('03/31/2019')) Returns registrations entered between those two dates.

INLIST() Lets you do a set of “OR” values, so if you wanted any Name that was in any one of three or four or five or ... cities.

Examples: INLIST(NMCITY,“Lincoln”,“Omaha”,“Ames”,“Dubuque”) gets a name if the city matches one of these four.

NOTE: Case does matter here. I.e.: if you are looking for a Course Character UDF where it might have been entered in different capitalizations, then use UPPER() on the match field and enter values in UPPERCASE.

```
INLIST(upper(CUDFC2),‘BLUE’,‘GREEN’,‘YELLOW’)
```

SUBSTR() Allows you to do a match for a section of a character or memo data field (particularly helpful in Course Code custom searching) The basic format is:

SUBSTR(<fieldname>,<starting character position>,<ending character position>)

The SM On line help guide has a detailed description of this (Search for SUBSTR)

Multiple Conditions: If you need to compare TWO or more different fields, you can use one of the concatenation commands (AND or OR). Just note that if you are using mixed commands (both AND's and OR's in the same condition) you may need to use Parenthesis.. to insure that your "Formula" returns what you want. (similar to using + and – along with X and / in a math formula). IF you are doing mixed bag conditions... you may want to consult with your technician.

Example= NMSTATE='NE' or NMORG='MANUF' (get everyone in Nebraska along with everyone with a Org code of MANUF) So a name with EITHER one of these codes will be pulled.

Example= NMSTATE='NE' AND NMORG='MANUF' (ONLY get people in Nebraska If they ALSO have the Org code of MANUF). BOTH must be true.

Some Case Study Examples:

F2 Quick List

Looking for courses with a CUDFC2 value = 'International' (and you are not sure how it was capitalized)

UPPER(cudfc2)='INTERNATIONAL'

Want NEW courses in current term:

COCRSE='19F' and CONEW

Want Courses added after August 1st 2019:

COADDATE > CTOD('08/01/2019')

F5 Name Finder:

Want to show names that were added within the last 14 days:

NMADDATE > DATE()-14

Quick list of names with interest code ='MARKET'

NAMECODE.NMCODE='MARKET'

Find names that have NO email address

EMPTY(nmemail)

Quick list of names with User Defined Field C2 that says Resident

`NAMEUDFS.NUDFC2='Resident'`

If you are NOT sure how the data entry user capitalized Resident

`UPPER(NAMEUDFS.NUDFC2)='RESIDENT'`

Looking for a specific word typed anywhere in Name Special Needs (and ignoring capitalization)

`"HEARING" $ UPPER(NMSPEC)`

F7 Pay Grabber

Finding payments with Deposit # = '190701' :

`PYACCTFOR='190701'`

Finding payments with the Pay User Character field 1 = Grant:

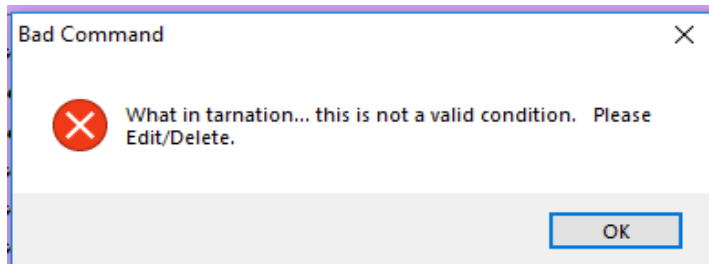
`PYUDFC1='Grant'`

If you're not sure about capitalization, use upper case.

`Upper(PYUDFC1)='GRANT'`

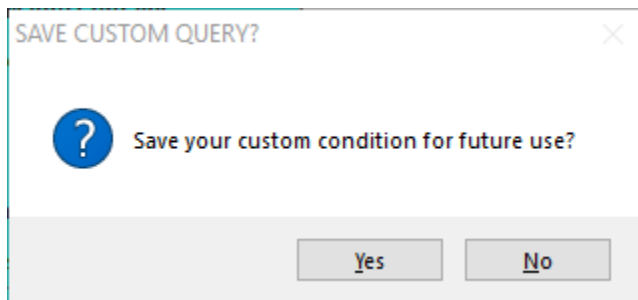
Forging ahead on your own:

As noted at the beginning, when doing the F - look up/search keys, there are no penalties for a mistake. You'd just get a message like:



And then you can try again.

Remember: You can ask Manager to Remember! On most custom conditions, after you've run a condition, you get a message:



So if this is a special code that you'd use again or would be an example that you could recall and edit, you may save it and you can recall it by hitting the dropdown arrow.

Summary:

Your Student Manager reporting system is very powerful, but sometimes it's nice to do quick pulls of information with the F keys. The Custom Condition option does offer you an unlimited ability to explore and report your data.