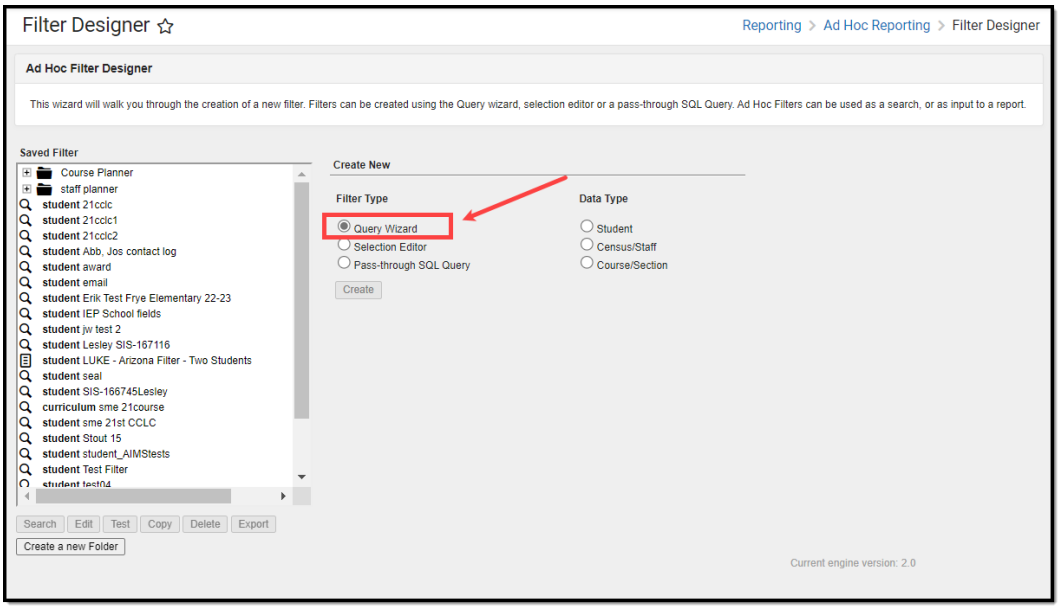
Query Wizard

Last Modified on 08/31/2026 1:53 pm CDT

Tool Search: Filter Designer

In the Query Wizard, elements are organized in a straightforward pattern, so it is easy to select the elements needed. Filters can be designed with student information, census/staff information or course/section information. Queries for students and course/section data pulls results from the calendar selected in the Campus toolbar. Census/Staff data pulls results from the entire Campus database, regardless of the calendar selected.



See the Filter Designer Tool Rights article for information about rights needed to use the Query Wizard.

See the Tool Rights article for more information about Tool Rights and how they function.

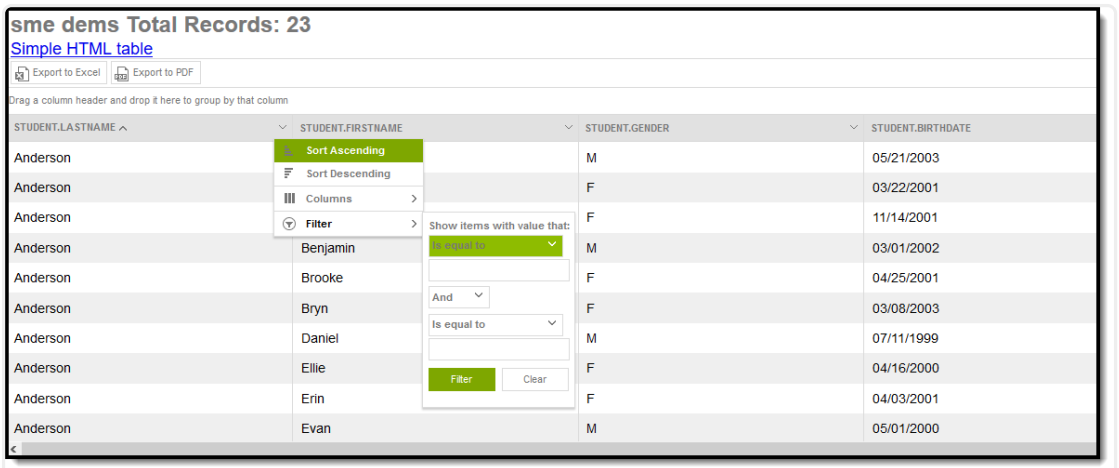
Unless using the Data Warehouse, queries should be created in such a way to avoid large results. Generating large queries may cause performance issues.

An ad hoc row limit is set on the database at 5 million rows. Any query that returns more than this is shortened. A warning message displays when this occurs.
When generating large queries and the Ad hoc Row Limit is met:

- Select fewer fields to include in the query
- Add more filters (see Functions) to reduce the number of records
- Use direct SQL access

Filters including GPA fields may task the server. It is recommended that these queries be generated after normal school hours.

Filters built in the Filter Designer display in HTML format. The HTML output allows for column sorting, filtering, grouping, and exporting to Excel or PDF.



HTML Filter Display

To view the output in a simple HTML table, click the link at the top of the output. This displays the output without the ability to sort, group and organize the columns.

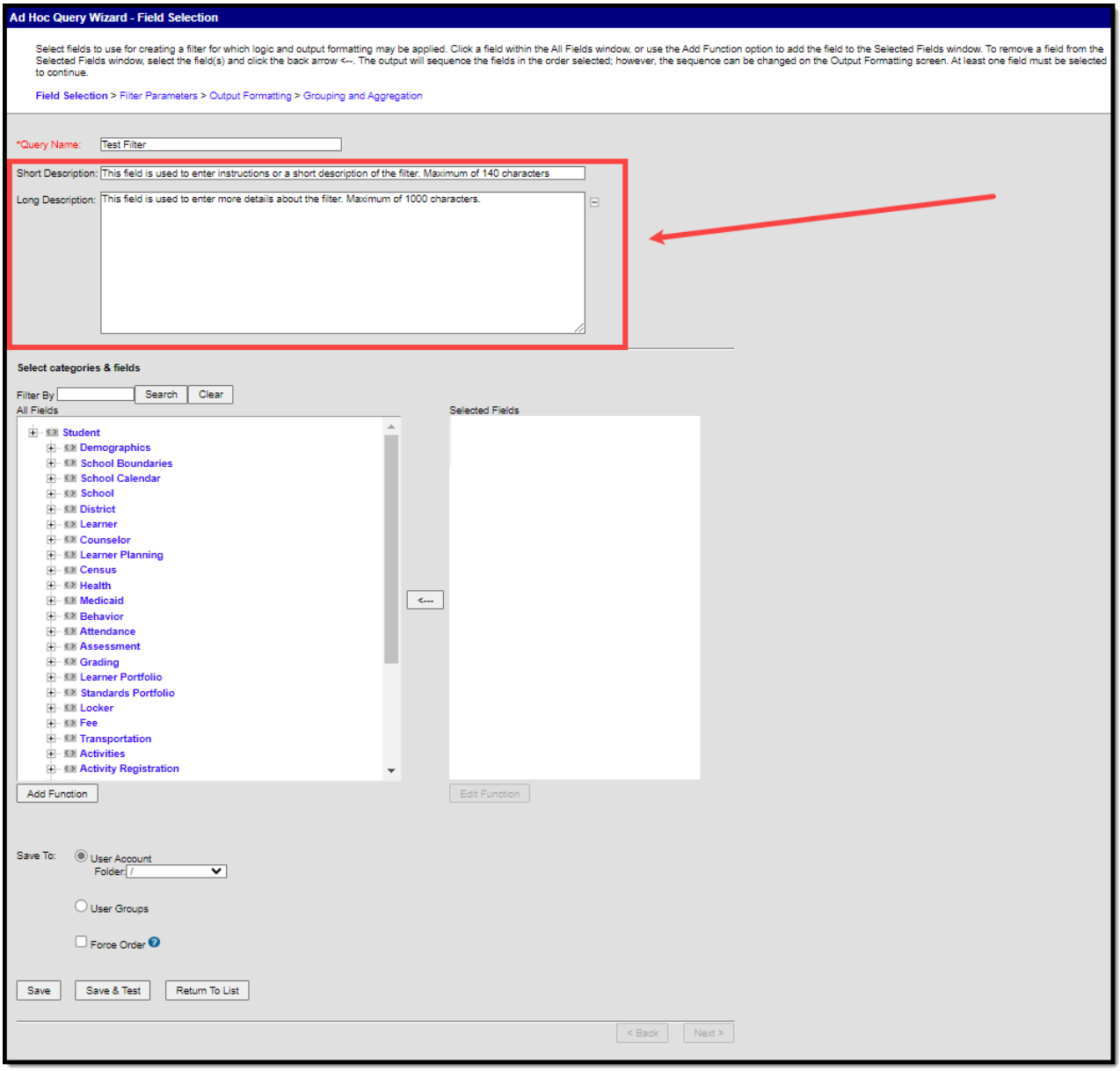
Query Wizard functionality allows users to easily create Ad hoc filters by organizing elements in a straightforward manner. Query Wizard filters are dynamic and always pull current information from the database based on the fields and filter options selected.

When using Custom Tab fields within Ad Hoc Query Wizard, all students are included in the results even if the student does not have a record within the custom dated tab. To exclude students without records for fields from a custom tab that is a Table or List Element tab type, set the statusDate Operator to **IS NOT NULL**. When pulling in fields from a custom tab that is a Table or List Element tab type, Ad Hoc logic outputs every possible combination based on a specific date and time. The Table Tab Type stores specific times. The List Element Tab type always stores 12:00 AM. See the Custom Tool Setup article for more information.

Query Wizard Features

Short and Long Filter Descriptions

This provides additional information and context about the filter. It's displayed when a user selects that filter from the Saved Filters list and when the filter is being modified.

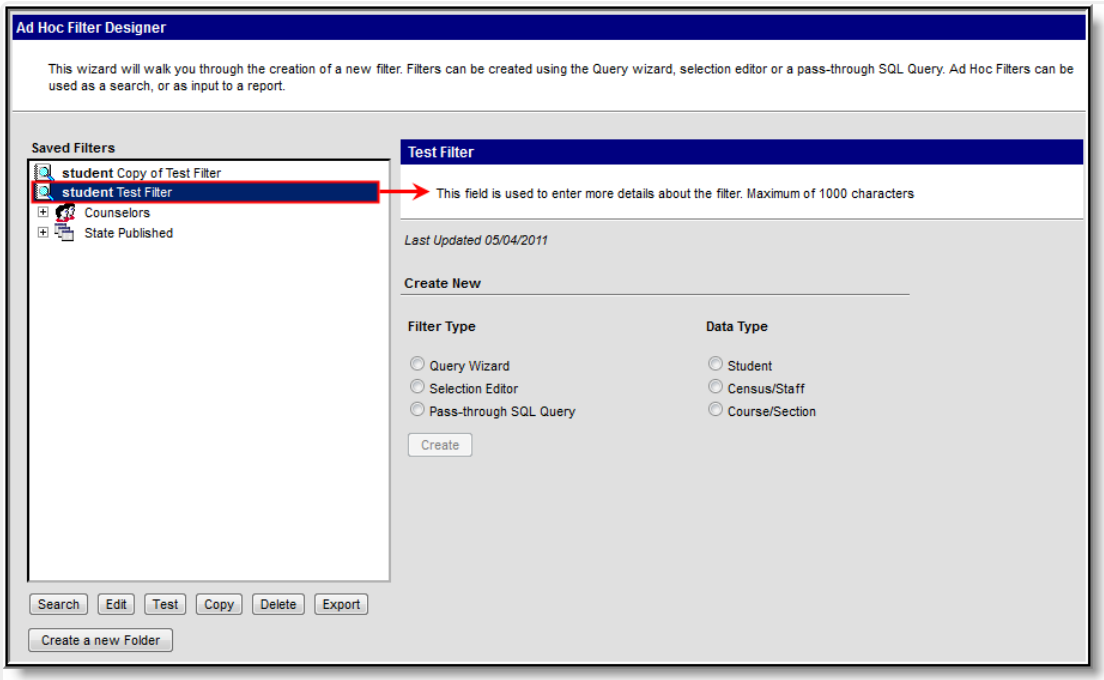


Click here to expand...

To attach short and/or long descriptions to a filter, enter this information within the **Short Description** and **Long Description** text fields. To access the **Long Description** text box, select the (**+**) icon. Once the filter itself is saved, all descriptions entered are saved.

Once a filter has a long description entered and saved, this information displays on the Filter Designer editor when the filter is selected in the **Saved Filters** window. This is useful when determining what filter to use as well as communicating any important information about the filter prior to editing or making modifications. If a short description has been entered, this information displays when the cursor hovers over the filter within the **Saved Filters** window.

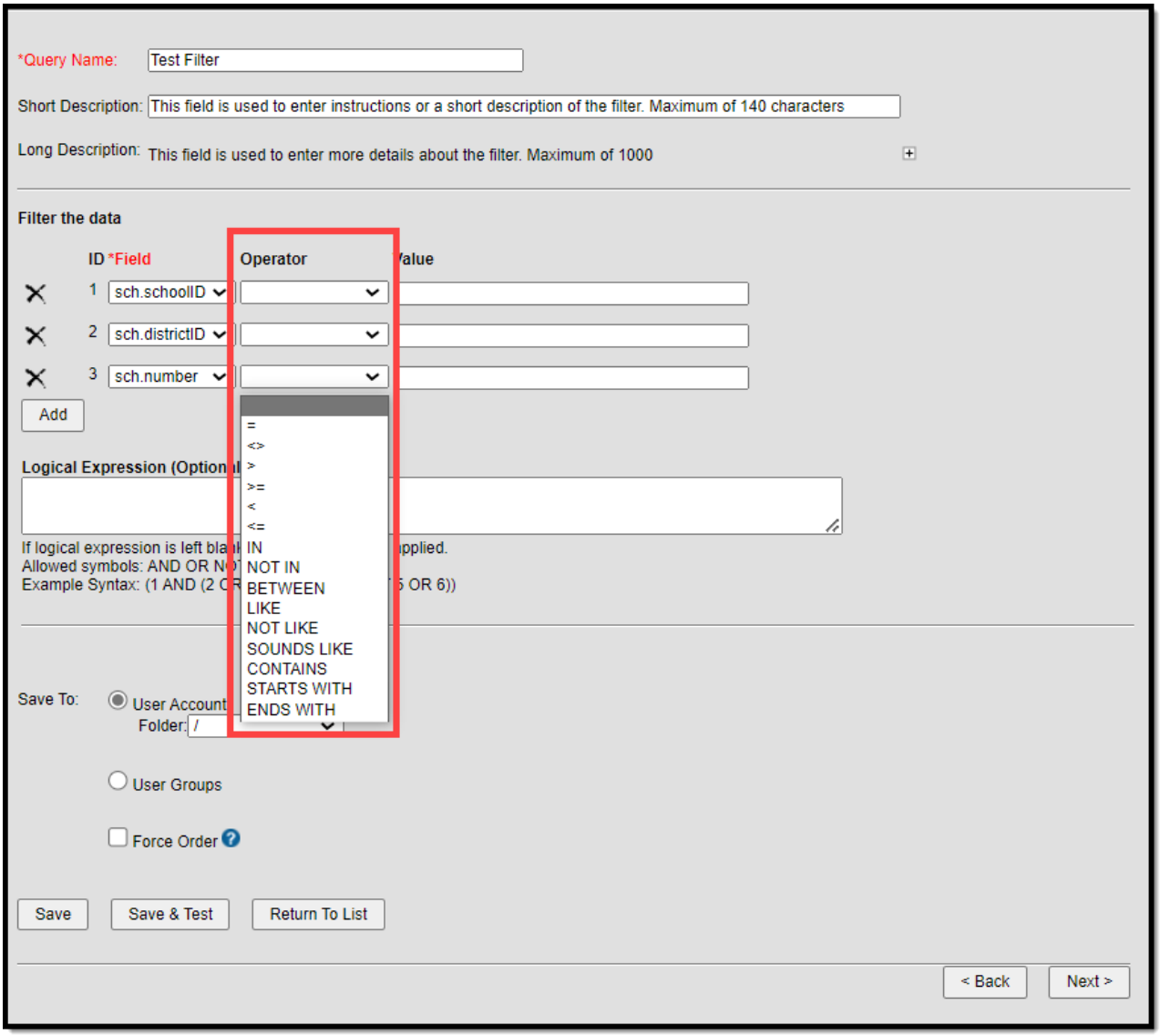
Both the Short and Long Descriptions display when a saved filter is edited/modified. Although the Long Description field appears locked, it can be modified by selecting the (**+**) icon.



Filter Description Display

Filter Operators

Filter operators allow users to set specific parameters per field within a filter. These parameters uniquely filter each field while maintaining the filter as a whole.



Users may apply multiple operators to the same field by clicking the **Add Filter** button and selecting a field. If a **Logical Expression** exists, all fields assigned an Operator must be included within the expression.

Click here to expand...

The following table describes each available filter operators:

Operator	Results	Example
= (Equals)	Returns exact match of value.	student.grade=3 Only students in grade 3 are returned.

Operator	Results	Example
< > (Does not equal)	Returns results not equal to the value.	student.gender < > M Students who have a Gender = F assigned on their Identities record or who do not have a value entered in the Grade field are returned. This operator allows NULL values.
> (Greater than)	Returns results that are greater than the entered numeric value.	student.age > 16 All students older than 16 years of age are returned.
> = (Greater than or equal to)	Returns results that are greater than or equal to the entered numeric value.	student.age >= 16 All students 16 years of age and older are returned.
< (Less than)	Returns results that are less than the entered numeric value.	student.age < 16 All students under the age of 16 are returned.
< = (Less than or equal to)	Returns results that are less than or equal to the entered numeric value.	student.age <= 16 All students 16 years of age and younger are returned.
IN	Includes value.	student.grade IN 9,10 All students in 9th and 10 grade are returned. When using this format, do not put spaces after the comma.

Operator	Results	Example
NOT IN	Excludes value.	student.grade NOT IN 11,12 All students not in 11th or 12th grade are returned. This operator allows NULL values. When using this format, do not put spaces after the comma.
BETWEEN	Filters data between two specified values. Works with numbers, dates and strings. If a date field is selected, the following options are available: • **DATE** - Returns data based on the specified date range (where the starting date is sub-option 1 and the ending date is sub-option 2). • **TODAY** - Filters data based on dates that occur from a specific date through today or vice versa. • **TOMORROW** - Filters data based on dates that occur from a specific date through tomorrow or vice versa. • **YESTERDAY** - Filters data based on dates that occur from a specific date through yesterday or vice versa. • **DAYS BEFORE** - Filters data based on the number of days (sub-option 1) prior to sub-option 2 through sub-option 2. • **MONTHS BEFORE** - Filters data based on the number of months (sub-option 1) prior to sub-option 2 through sub-option 2. • **DAYS AFTER** - Filters data based on sub-option 1 through the number of days (sub-option	For **BETWEEN**: student.stateID BETWEEN 00001 THROUGH 100000. All students with a State ID between 00001 - 100000 are returned. For **DATE**: student.birthDate BETWEEN DATE 10151995 THROUGH DATE 10152010. All students with a birth date between 10/15/1995 - 10/15/2010 are returned. For **TODAY**: student.startDate BETWEEN TODAY THROUGH TODAY. All students who began an enrollment in the school today (current date) are returned. For **YESTERDAY**: student.startDate BETWEEN YESTERDAY THROUGH DATE 10152010. All students who began an enrollment in the school yesterday through 10/15/2010 are returned. For **DAYS BEFORE**: student.startDate BETWEEN DAYS BEFORE 4 THROUGH YESTERDAY. All students who began an enrollment in the school 4 days before yesterday through yesterday are returned. For **MONTHS BEFORE**: student.startDate

Operator	Results	Example
	2) after the sub-option 1 date. • **MONTHS AFTER** - Filters data based on sub-option 1 through the number of months (sub-option 2) after the sub-option 1 date.	BETWEEN MONTHS BEFORE 5 THROUGH TODAY. All students who began an enrollment in the school 5 months prior to today through today are returned. For **DAYS AFTER**: student.startDate BETWEEN DATE 10152010 THROUGH DAYS AFTER 5. All students who began an enrollment in the school on 10/15/2010 through 10/20/2010 (5 days after) are returned. For **MONTHS AFTER**: student.startDate BETWEEN DATE 10152010 THROUGH MONTHS AFTER 5. All students who began enrolling in the school on 10/15/2010 through 3/15/2011 (five months after) are returned.
IS CURRENT USER	Returns the current user's ID.	**Learner Plan Manager** Setting learningPlan.planManagerPersonID IS CURRENT USER reports the current user's ID, along with data only applicable to that user. **Current Teacher Sections** For courseSection.personID IS CURRENT USER limits the results to students in the current teacher's section. This is useful for a report of student birthdays with a homeroom, or a Spirit Squad Advisor who needs to make locker signs and needs a list of participants and locker information.
LIKE	Searches for test string in the field.	course LIKE hist All courses like History 101 are returned.
NOT LIKE	Searches for test string and filters data that is not like the user-defined value.	course NOT LIKE hist All courses not like Hist are returned. This operator allows NULL values.

Operator	Results	Example
SOUNDS LIKE	Uses a database function to return names with similar sound patterns.	student.lastName SOUNDS LIKE Ball Names such as "Ball," "Bell" and "Boll" are returned.
CONTAINS	Searches for strings that include the same data entered by the user in the field. Any string that does not contain the user-defined value is filtered out. Any wildcard characters entered are treated as standard SQL wildcards.	student.birthCountry CONTAINS Cana All students with a Birth Country that contains "Cana" are returned.
STARTS WITH	Searches for strings that begin with the same data entered by the user in the field. Any string that does not contain the user-defined value is filtered out. Any wildcard characters entered are treated as standard SQL wildcards.	student.birthCountry STARTS WITH Mexi All students with a Birth Country that begins with "Mexi" are returned.
ENDS WITH	Searches for strings that end with the same data entered by the user in the field. Any string that does not contain the user-defined value is filtered out. Any wildcard characters entered are treated as standard SQL wildcards.	student.birthCountry ENDS WITH many All students with a Birth Country that ends with "many" are returned.
IS NULL	Returns fields that are completely NULL (0 is considered a value).	student.stateID IS NULL All students who do not have a state ID are returned.
IS NOT NULL	Returns all fields that are not NULL (0 is considered a value).	student.ssn IS NOT NULL All students who have a stateID are returned.
IS TODAY	Returns result dates as the current date.	start.date IS TODAY Entries where the start.date is the current date are returned.
IS YESTERDAY	Returns result dates as of yesterday's date.	start.date IS YESTERDAY Results for one day previous to the current date are returned.

Operator	Results	Example
IS TOMORROW	Returns result dates as of tomorrow's date.	end.date IS TOMORROW Results for one day after the current date are returned.
IN THE MONTH	Returns all database field data for the month entered. This operator allows both numbered dates and spelled-out dates (e.g., 10 or October). It also allows for both upper and lower case letters. If spelling out a month, users must enter at least the first three characters (e.g., Oct for October).	employment.districtStartDate IN THE MONTH October All employees who have a district employment Start Date within the month of October are returned. This operator does not look at the Year or Calendar selected in the Campus toolbar. All historical and current district employment records with a Start Date in October are returned.
=TRUE	Returns checkbox values of "true" (checkbox is marked)	enrollment.stateExclude = TRUE All students with the State Exclude checkbox marked on their enrollment records are returned.
=FALSE	Returns checkbox values of "false" (checkbox is not marked)	enrollment.stateExclude = FALSE All students who do not have the State Exclude checkbox marked on their enrollment records are returned.

In addition to the options above, wildcard searching is also available. The following is a list of options:

Wildcard or Pattern	SQL Meaning	Standard Examples
%	0 or more characters	Entering the word *Man* returns the same results when entering *Man%*. *%son* finds names that end in *-son*: Johnson, Manson, Jason-Benson, etc.
_ (underscore)	One character	*Olson_Zierke* and *Olson Sierke* return the same results. *L__* (with two underscores) does not look only for 3-character names that start with *L*, but _L_e_ finds names where *L* is the first and *e* the third character (e.g. *Lee, Luewenhook*). If the three underscores are entered at the end of a name, like *Dan___*, results list names with three additional letters (*Daniel*).

Wildcard or Pattern	SQL Meaning	Standard Examples
[token]	A range of possible characters	*L[ae]* finds names that start with *La* or *Le*.
,James	No SQL wildcard	Searches for first name equal to or beginning with James. This can only be used in the Quick Search fields.
Gonzales-Uribe	Compound name	Finds that last name. This returns compound names regardless of whether they are linked by a space or hyphen.
Gonzales Uribe or **Gonzales_uribe** or **Gonzales%uribe**	A compound name with a space.	Finds the name with or without a space or hyphen. Try wildcards if there is a space between the compound names.

Users can also use the following combinations when using the *Like* operator:

Wildcard or Pattern	SQL Meaning	Standard Examples
%	0 or more characters	L% finds names that start with L L finds names that contain an L LAN finds names containing LAN (Blanko, Landesburg, Blankenship, etc.)
_ (underscore)	One character	*L__* (two underscores) finds *Lee* and *Lor*, not *Luewenhook*.
[token]	A range of possible characters	*L[ae]%* finds names that start with *La* or *Le.*
^	Negation of token	*L[Query Wizard^ae]* finds names that do not start with *La* or *Le*.

Rules for Operators by Data Type

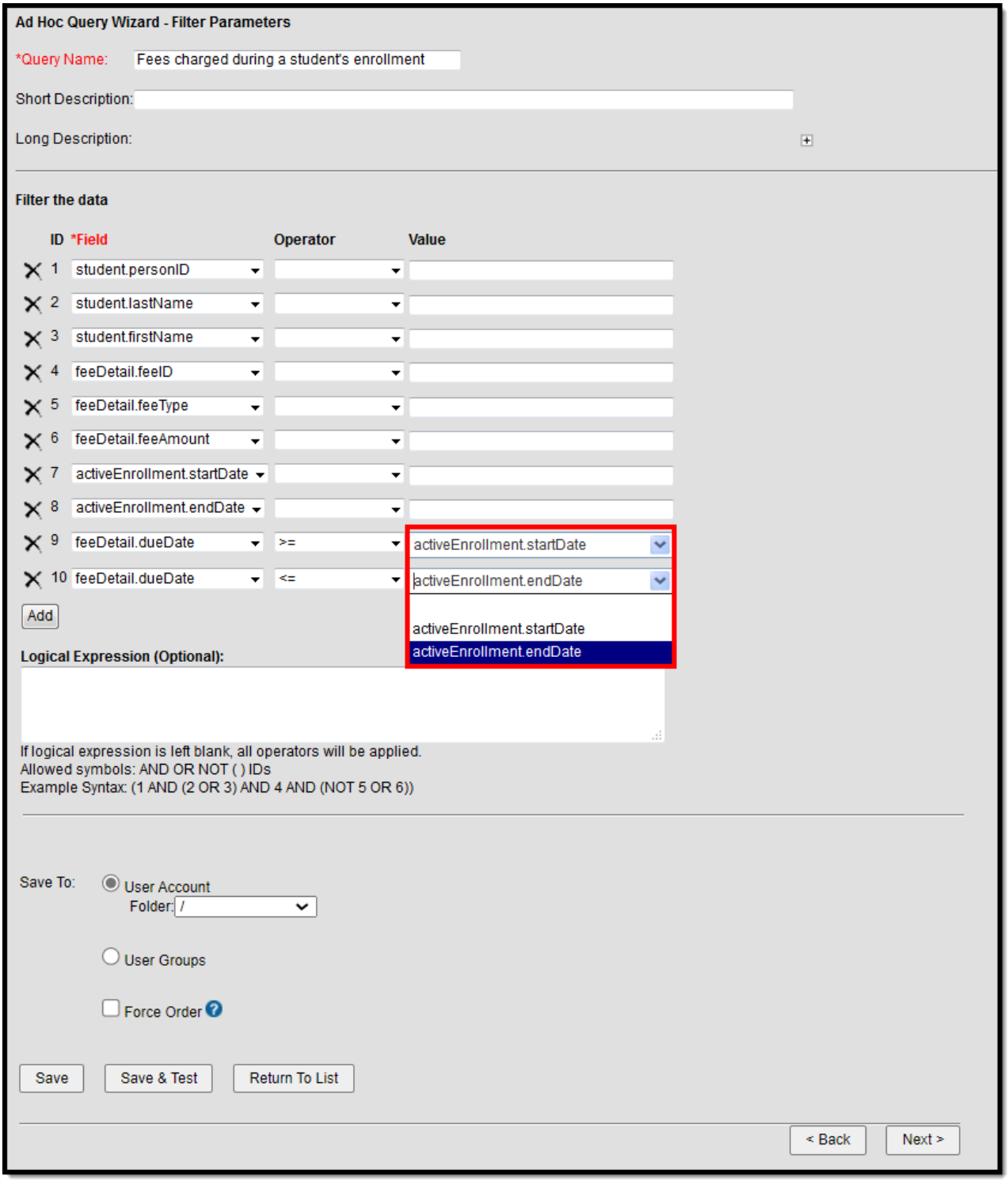
The following table describes all rules for allowing or disallowing operators by data type, where Y = Allowed, N = Not Allowed, and D = Depends on Field.

Option	Number	Float	String	Date	Text	Bit
>	Y	Y	Y	Y	Y	N

Option	Number	Float	String	Date	Text	Bit
>=	Y	Y	Y	Y	Y	N
<	Y	Y	Y	Y	Y	N
<=	Y	Y	Y	Y	Y	N
< >	Y	Y	Y	Y	Y	N
=	Y	Y	Y	Y	Y	N
IS NULL	D	D	D	D	D	N
IS NOT NULL	D	D	D	D	D	N
BETWEEN	Y	Y	Y	Y	Y	N
IS TODAY	N	N	N	Y	N	N
IS YESTERDAY	N	N	N	Y	N	N
IS TOMORROW	N	N	N	Y	N	N
IN	Y	Y	Y	Y	Y	N
NOT IN	Y	Y	Y	Y	Y	N
LIKE	N	N	Y	N	N	N
STARTS WITH	N	N	Y	N	N	N
ENDS WITH	N	N	Y	N	N	N
CONTAINS	N	N	Y	N	N	N
SOUNDS LIKE	N	N	Y	N	N	N
=TRUE	N	N	N	N	N	Y
=FALSE	N	N	N	N	N	Y

Use a Field as an Operator Value

Depending on the operator chosen for the field, a field may be used as an operator's value allowing a comparison between two fields. Logic only allows fields of the same data type to be used as the Operator's Value. For example, date fields are allowed to use other date fields as an operator value. When the appropriate operator is used, the Value column can act as a dropdown list while remaining static allowing the user to select a field or input a value. Deleting a field also removes it from the Value field, clearing out the operator for the field using it. Additionally, replacing a field with the Element Replacement tool replaces the field and the operator's value if the replaced field is being used as the value.



In the above example, a query was set to report students with fees charged during enrollment. Using the fields activeEnrollment.startDate (7) and activeEnrollment.endDate (8) as operator values for feeDetail.dueDate (9, 10), the query reports students with fees due on or after the student's active enrollment start date AND on or before the student's active enrollment end date.

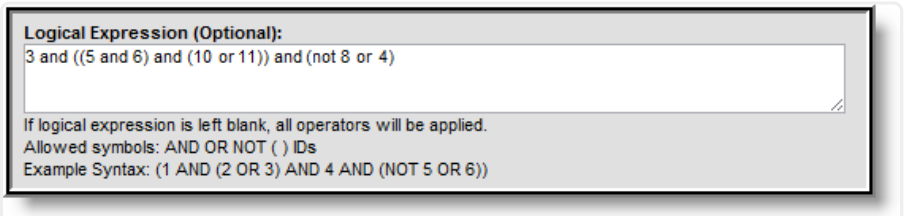
Operators Allowed to Use a Field as Values

Operator	Allowed
>	Y
>=	Y
<	Y
<=	Y
< >	Y
=	Y
BETWEEN	Y
IS CURRENT USER	N
IN THE MONTH OF	N
MONTHS BEFORE	N
DAYS BEFORE	N
IS NOT NULL	N
IS NULL	N
IS TODAY	N
IS YESTERDAY	N
IS TOMORROW	N
IN	N
NOT IN	N
LIKE	N
STARTS WITH	N
ENDS WITH	N
CONTAINS	N
SOUNDS LIKE	N
=TRUE	N
=FALSE	N

Logical Expressions

The Logical Expression field allows users to incorporate conditions between fields within a filter. This field effectively uses the OR, AND, and NOT conditions between fields and groups of fields.

- Only fields assigned an **Operator** are allowed to be included within logical expressions.
- Logical Expressions are created using the ID number associated with each field.



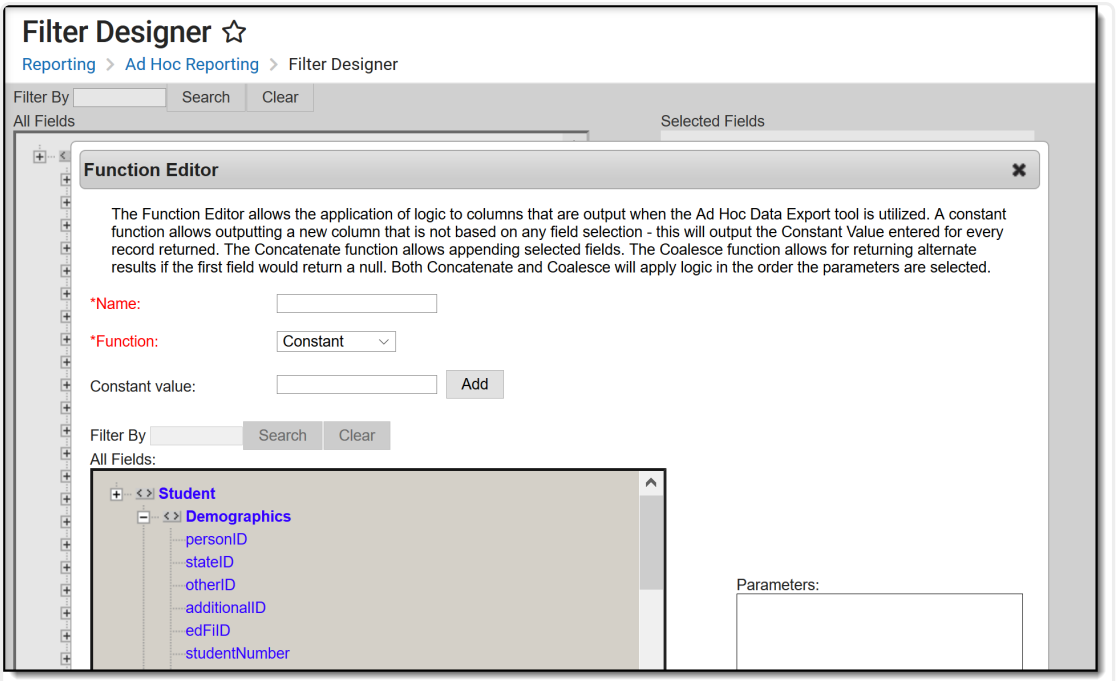
Logical Expression with a Filter

Logical expressions can be grouped using ( ) symbols and the ID number to define the order in which the tool should include or exclude a person. In the example above, the ( ) symbols indicate the tool should determine the student's End Date (5) and grade (6) and include these students depending on whether they are Asian (10) or White (11). This determination and group of students is then applied to the remaining parts of the logical expression.

Using ( ) symbols is especially useful when using the OR condition, as users can include or exclude people based on whether or not they meet the criteria for the fields included within a group of fields. For example, students with a State ID less than 1000 (8) or an End Status populated (4) are not included in the remaining calculation for the logical expression.

Functions

Functions can be added to filters, which allow logic to be applied to field columns when the filter is generated via the Data Export tool. To add a function to a filter, select the **Add Function** button. The **Function Editor** appears in a new window.

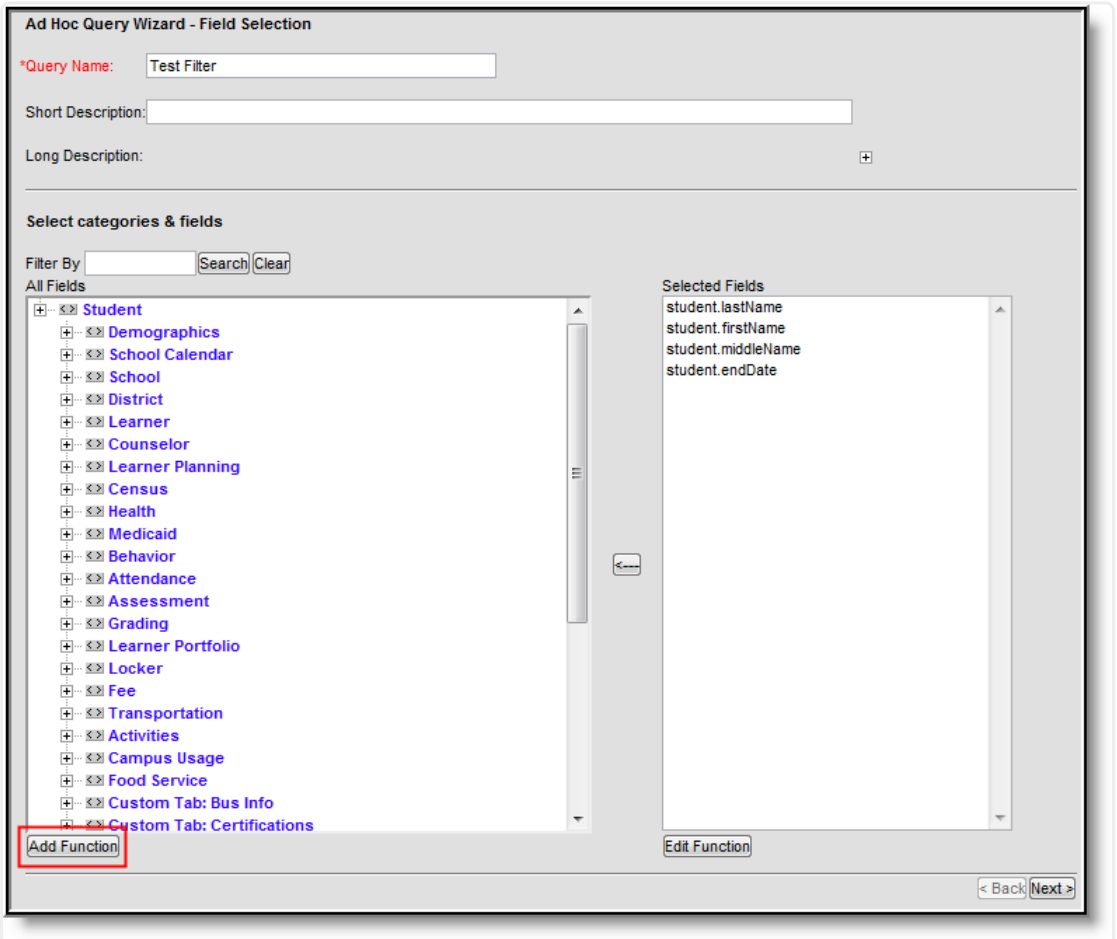


Add Function

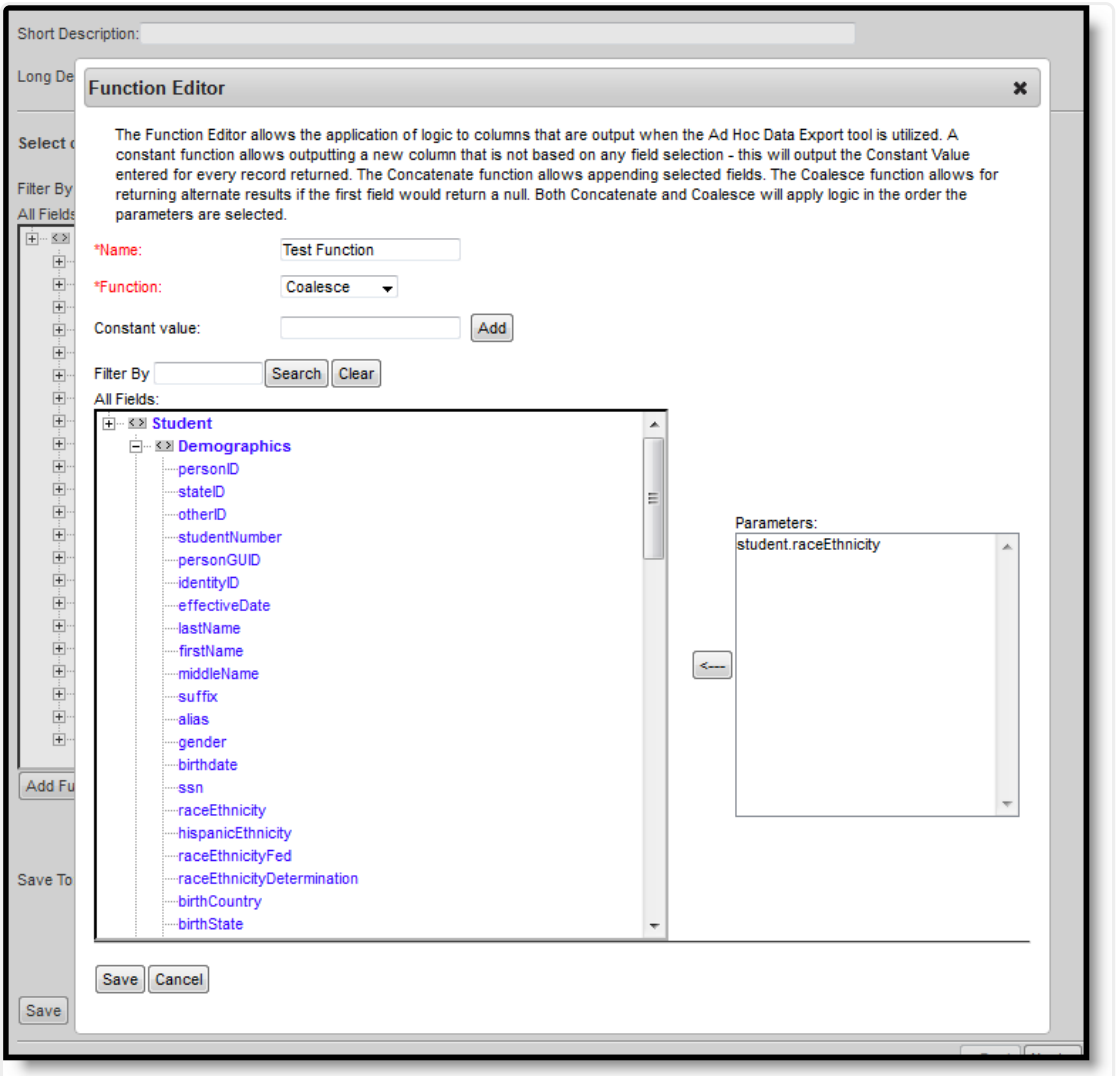
Click here to expand...

Add Functions to Queries

1. Enter the **Name** of the function. This name differentiates the function from other functions within the **Selected Fields** window and on filters generated via the Data Export tool.
2. Select the desired **Function** from the dropdown list. The Function Descriptions section below provides descriptions and examples of each function. Once a function is selected, the **Filter By Search** field (see step 4) becomes active.
3. If the Constant **Function** is selected, enter the **Constant Value** and click the **Add** button. The value entered displays in the Parameters window and is reported on every record returned.
4. Use the **Filter By Search** field to search for desired fields. Entering a search value and clicking the **Search** button resets the list of fields to only return matching fields. Click the **Clear** button to remove entered search values and see the entire list of fields.
5. Select which fields to include within the function by clicking on each field within the **All Fields** window. Selected fields move into the **Parameters** window, indicating which fields have been added to the function.
6. Select the **Save** icon.

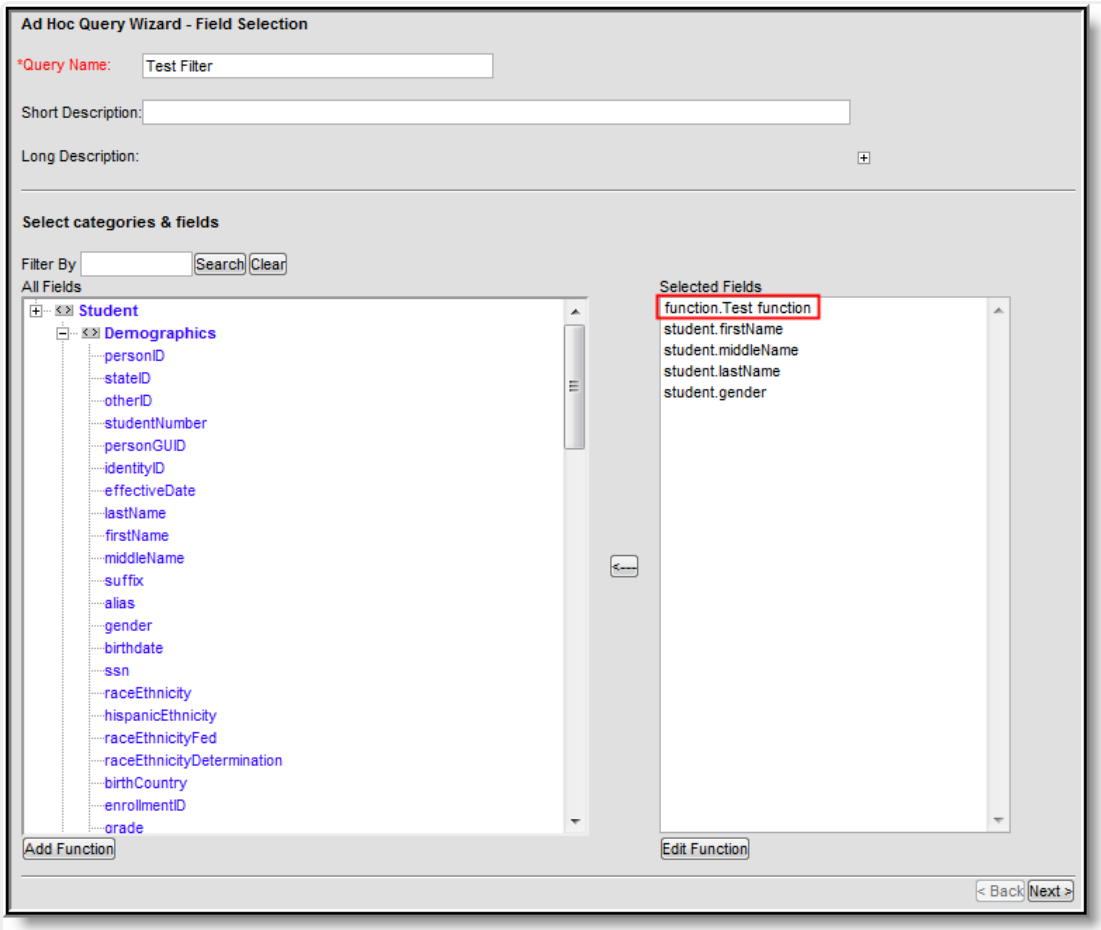


Adding a Function to a Filter



Enter Data into the Function Editor

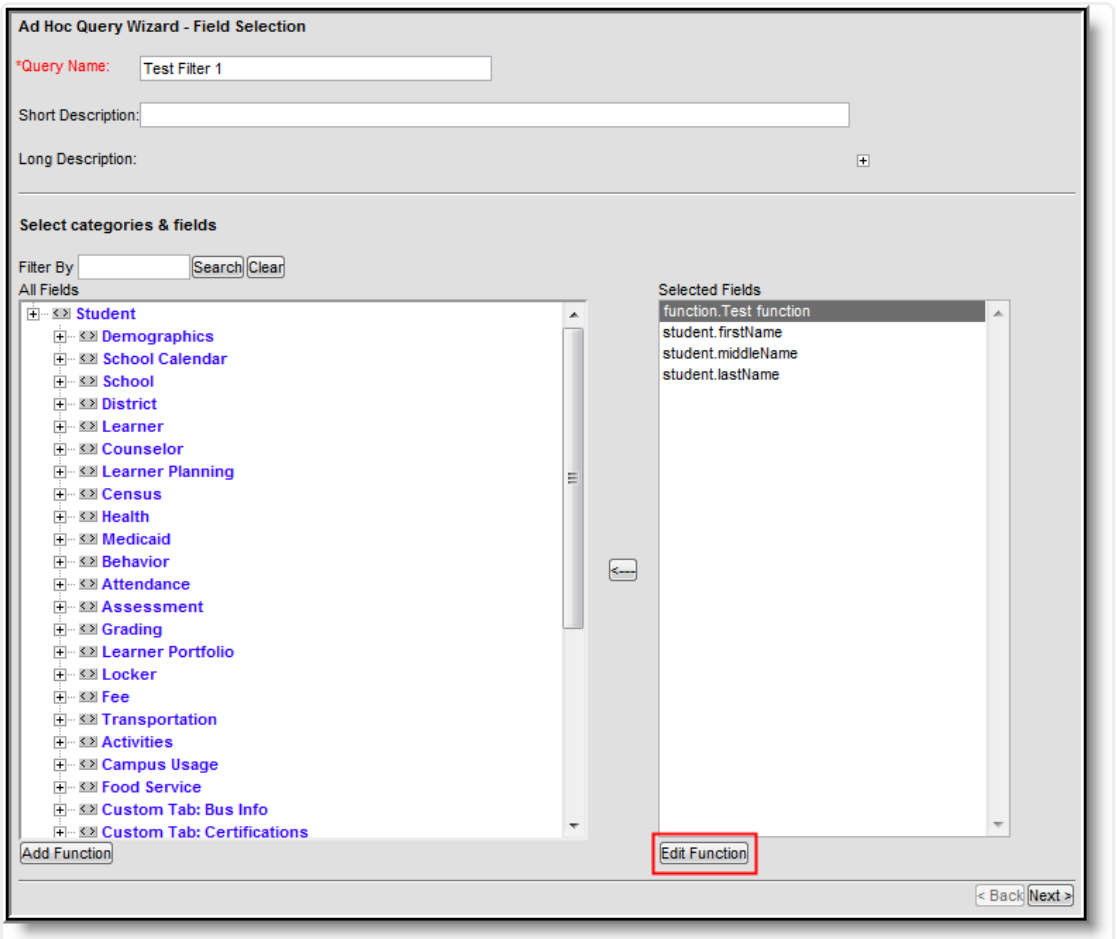
The **Field Selection** editor appears after saving the added function. Functions created and added to the filter are displayed in the Selected Fields window. The function's name always appears to the left of the period (*i.e.*, function.functionName).



View Functions Added to a Filter

Edit Functions

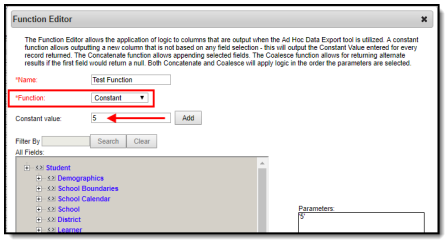
Existing functions can be edited by selecting the function within the **Selected Fields** window and clicking the **Edit Function** button.

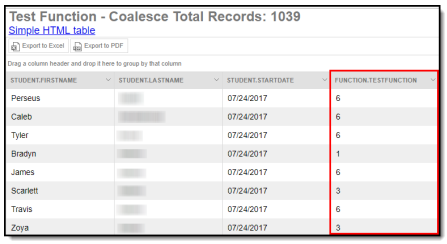
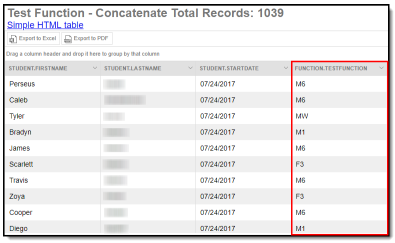


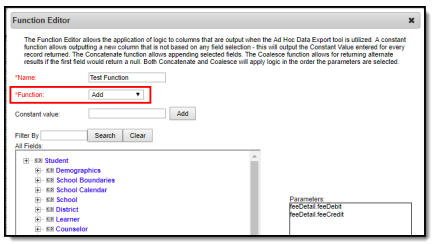
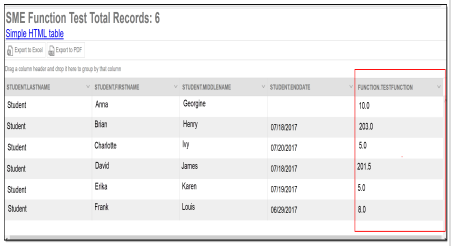
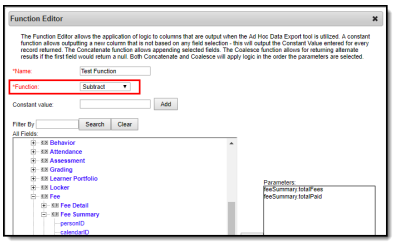
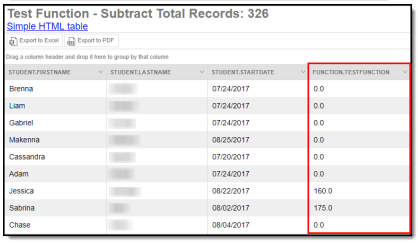
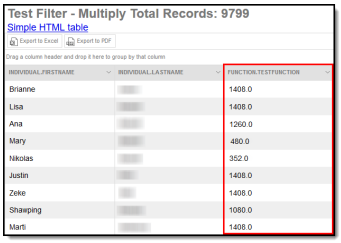
Edit Existing Functions

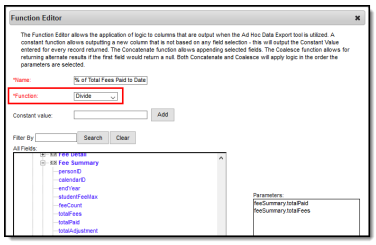
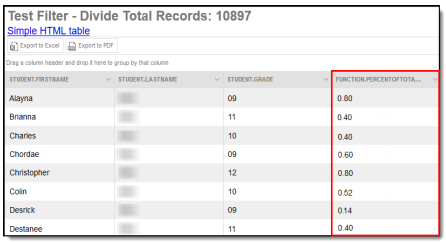
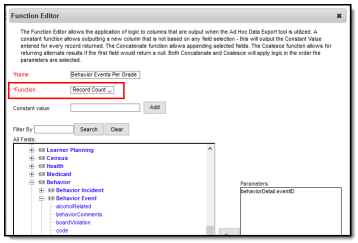
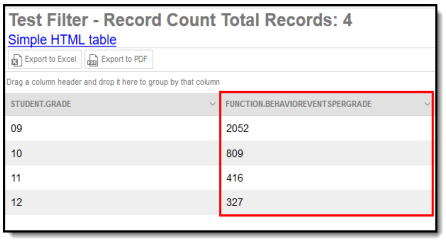
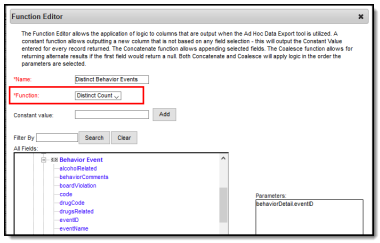
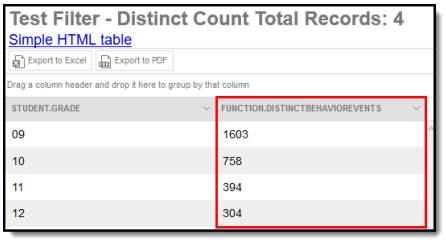
Function Descriptions

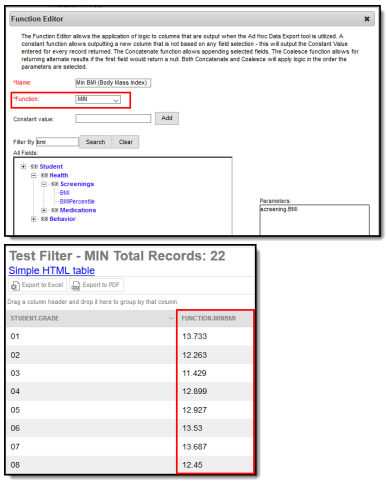
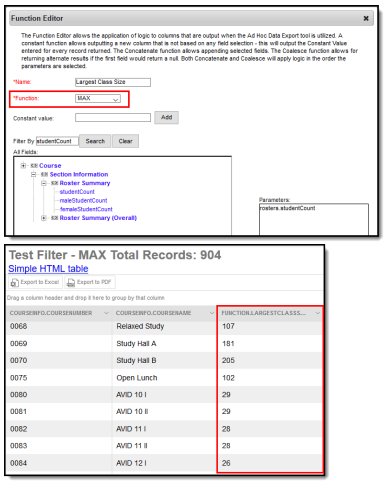
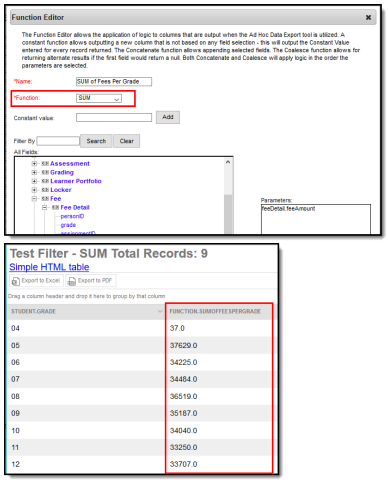
The following describes each available function.

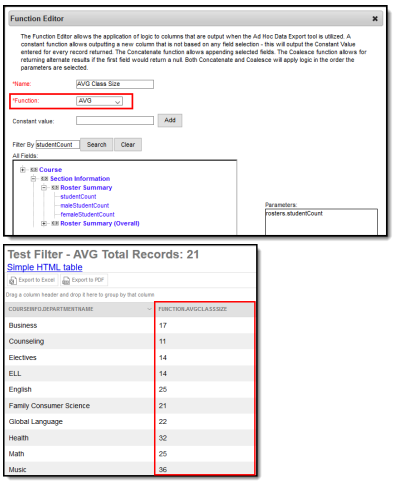
Function	Description	Example
Constant	The Constant function outputs the Constant Value entered on each record returned when the filter is exported. A Constant Value of 5 is entered and added to the filter in the examples to the right. When the filter is exported, a column is reported displaying the **Constant Value** entered.	

Function	Description	Example
Coalesce	The Coalesce function allows users to define multiple fields where logic pulls the first field, and if NULL, the second field is pulled, and so on down the line of added fields until a value is found. Logic pulls field values in the order fields are selected in the **Function Editor**. A Coalesce function for Federal Race Ethnicity and Race Ethnicity Determination fields was added to the examples to the right. This means logic first pulls and reports student Federal Race Ethnicity field values and for any that are NULL, the student's Race Ethnicity Determination reports. When the filter is exported, the function reports field data within a specific column. Student Federal Race Ethnicity values are reported.	 
Concatenate	The Concatenate function allows field values to be appended when the filter is exported. In the example to the right, a Concatenate function for fields Gender and Race Ethnicity was added. When the filter is exported, field values are appended and reported. Student Gender values (M, F) are reported alongside student Race Ethnicity values.	 

Function	Description	Example
Add	The Add function allows field values to be added together to output a single result (i.e., field 1 + field 2). In the example to the right, Fee Debit is added to Fee Credit to generate a total balance. When the filter is exported, field values are added and reported as a single value.	 
Subtract	The Subtract function allows field values to be subtracted from each other to output a single result. In the example to the right, total Fees are subtracted from Total Paid to report a student's outstanding balance. When the filter is exported, field values are subtracted and reported as a single value.	 
Multiply	The Multiply function allows field values to be multiplied together to output a single result (i.e., field 1 x field 2). In the right-hand example, employee hours per day are multiplied by the number of days employed for the year. When the filter is exported, field values are multiplied and reported as a single value.	 

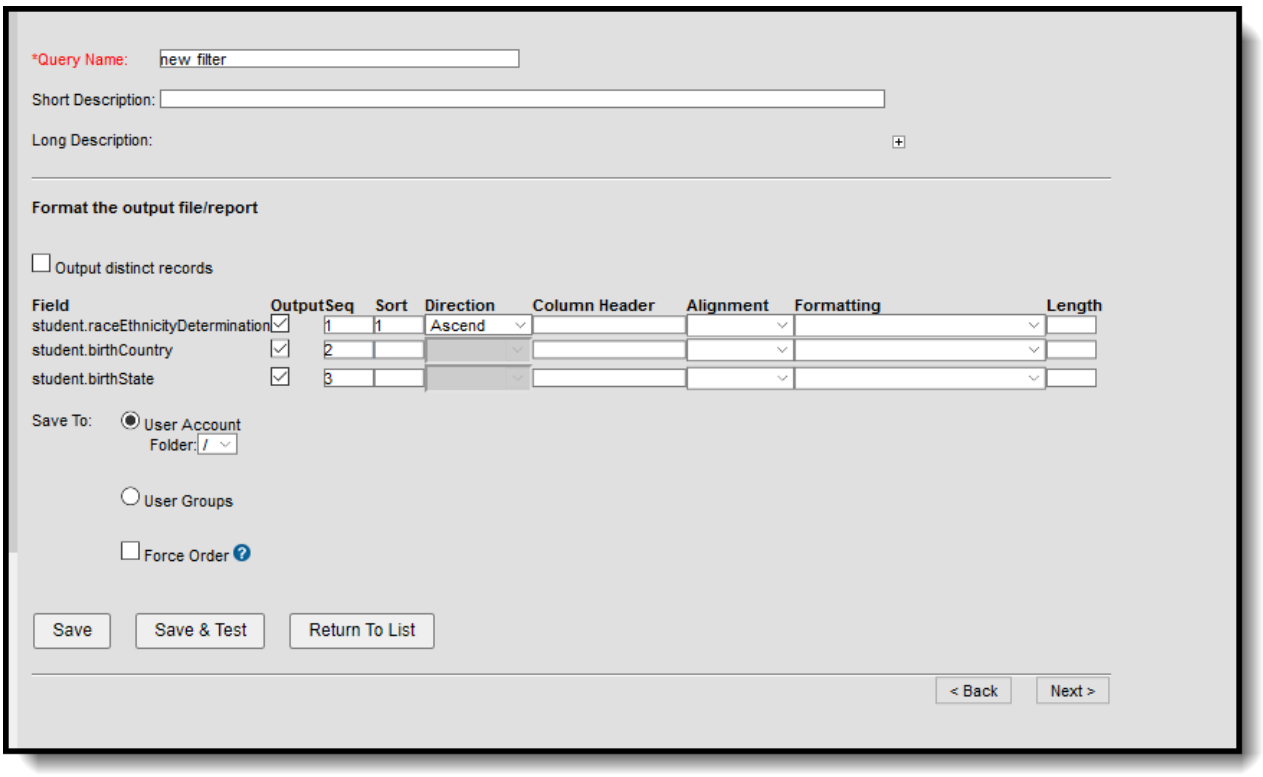
Function	Description	Example
Divide	The Divide function allows a field or more fields to be divided and output into a single result (i.e., field 1 / field 2). In the example to the right, the total number of fees is divided by the total amount of fees paid to get the percentage of total fees paid to date. If applicable, decimal places are included in the output. When the filter is exported, field values are divided and reported as a single value.	  Test Filter - Divide Total Records: 10897 Simple HTML table Export to Excel Export to PDF Drag a column header and drop it here to group by that column STUDENT.FIRSTNAME / STUDENT.LASTNAME / STUDENT.GRADE / FUNCTION.PERCENTOFTOTA... Alayna 09 0.80 Brianna 11 0.40 Charles 10 0.40 Chordae 09 0.60 Christopher 12 0.80 Colin 10 0.52 Desrick 09 0.14 Destanee 11 0.40
Record Count	The Record Count function allows users to report a record count for the field selected. In the example to the right, a record count of behavior events is used to report a count of behavior events per grade level. When the filter is exported, a record count of the field is calculated and reported.	  Test Filter - Record Count Total Records: 4 Simple HTML table Export to Excel Export to PDF Drag a column header and drop it here to group by that column STUDENT.GRADE / FUNCTION.BEHAVIOREVENTSPERGRADE 09 2052 10 809 11 416 12 327
Distinct Count	The Distinct function allows users to report a distinct count for the field selected. In the example to the right, a distinct count of behavior events is used to report the distinct count of behavior events per grade level. When the filter is exported, a record count of the field is calculated and reported	  Test Filter - Distinct Count Total Records: 4 Simple HTML table Export to Excel Export to PDF Drag a column header and drop it here to group by that column STUDENT.GRADE / FUNCTION.DISTINCTBEHAVIOREVENTS 09 1603 10 758 11 394 12 304

Function	Description	Example
MIN	The MIN function allows users to report the minimum value for a field. In the example to the right, the MIN BMI function reports the minimum BMI (Body Mass Index) per grade level. When the filter is exported, the MIN of the field is calculated and reported.	
MAX	The MAX function allows users to report the maximum value for a field. In the example to the right, the MAX student count is used as the function to report the largest class size per course. When the filter is exported, the MAX of the field is calculated and reported.	
SUM	The SUM function adds the value or field selected over all other aggregated fields. In the example to the right, the SUM of fee amounts is used to report the SUM of fees per grade. When the filter is exported, the SUM field is calculated and reported.	

Function	Description	Example
AVG	The AVG function allows users to report the average value for a field. In the example to the right, the AVG of roster student count is used to report the average class size per department. When the filter is exported, the AVG field is calculated and reported.	

Output Formatting

The Output Formatting editor lets users control how each field is reported and displayed when exported.



Click here to expand...

Output Formatting Descriptions

Field	Description
Output distinct records	If marked, data is outputted in unduplicated records based on field values. The following is an example of a filter containing student first name, last name, grade, gender, and behavior event type: • If a student has three behavior events for the same behavior event type and the Output distinct records checkbox is not marked, the student reports three records. • If the Output distinct records checkbox is marked, the same student now only reports one record.
Field	Fields selected from the All Fields window in the previous screen.
Output	This checkbox determines whether or not the field is included in outputted data. Deselecting this checkbox means data is still filtered and reported for this field and operators but not included in the output.
Seq	This field determines the sequence of outputted data.
Sort	This field determines the sort order of outputted field data.
Direction	This field determines if data is sorted ascending or descending. This field is only available if a value is entered in the Sort field.
Column Header	This field determines what header is displays for the field on exported files. Users are encouraged to enter a logical and easily identifiable column header for each field, as leaving the field blank results in the field name (i.e., student.stateID) being reported.
Alignment	The field determines how field data is aligned on files exported. Available options include: Left, Center and Right.
Formatting	The field determines how values are reported for the field when used in reports and exported files. Formatting options are important for filters used with reports which require specific formatting in order for the file to be correctly submitted to an entity or system. The following formatting options are available: • **Zero Pad** - numbers are padded with zeros to the left (*i.e.,* 444 zero padded becomes 000444) • **Space Fill** - values are filled with spaces in order to reach required field length • **Upper Case** - values are reported entirely in uppercase (i.e., Course is reported COURSE). This option is only available for text, char and varchar fields. • **Lower Case** - values are reported entirely in lowercase (i.e., Course is reported course). This option is only available for text, char and varchar fields. • **MM/DD/YYYY** • **MM-DD-YYYY** • **MMDDYYYY** • **YYYY/MM/DD** • **YYYY-MM-DD**

Field	Description

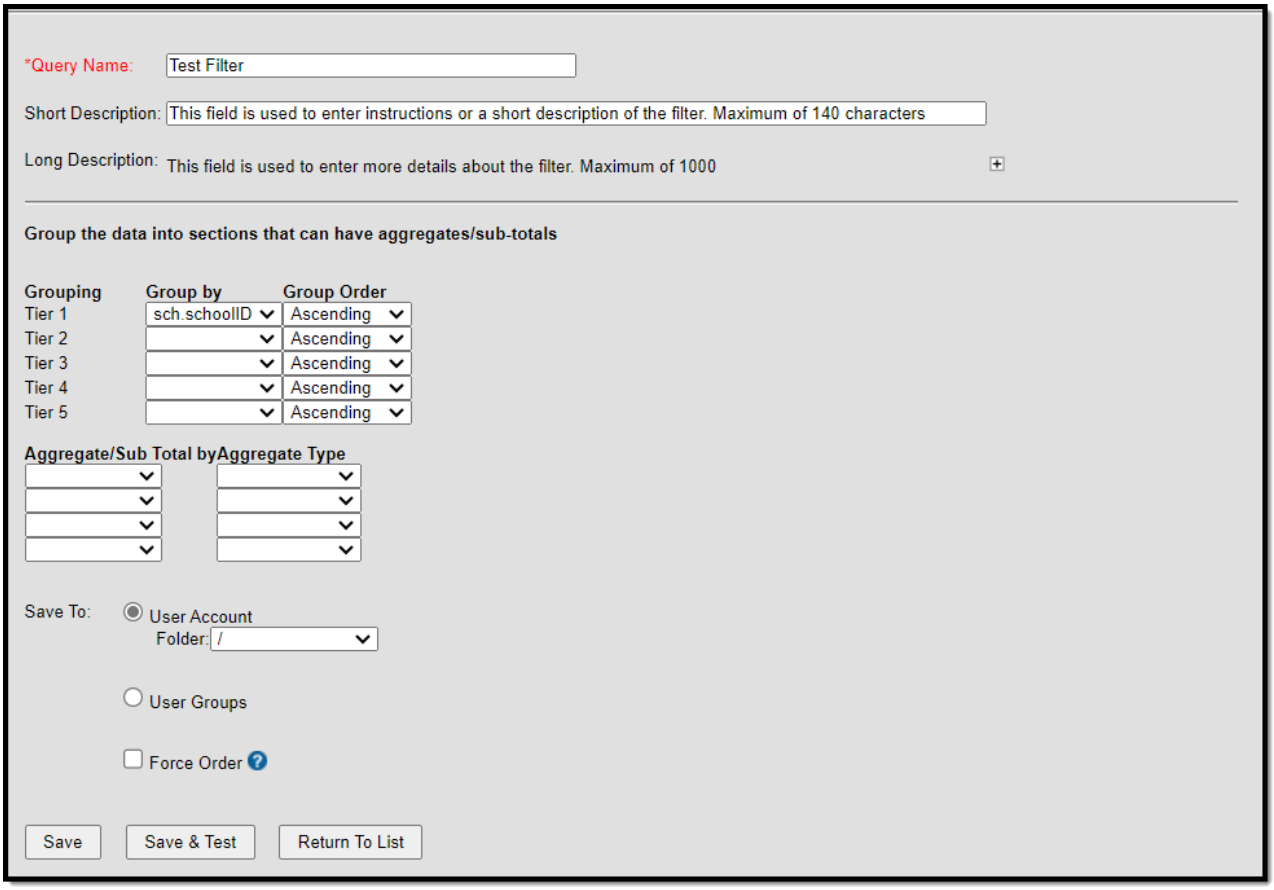
	• **YYYYMMDD**
	• **YYYY**
	• **YYYY/MM**
	• **YYYY-MM**
	• **YYYYMM**
	• **MM/YYYY**
	• **MM-YYYY**
	• **MMYYYY**
	• **MM/DD/YYYY hh:mm AM**
	• **MM-DD-YYYY hh:mm AM**
	• **YYYYMMDDHHmm** - This is similar to military time (e.g., 1:00 PM is 1300) because there is no AM/PM.
	• **Number Formatting Patterns:** Often used in accounting, finance, or spreadsheet applications to control how positive and negative numbers appear.
	  ◦ **1,234.5; - 1,234.5**
	  ◦ **1,234.5; (1,234.5)**
	  ◦ **$1,234.00; -$1,234.00**
	  ◦ **$1,234.00; ($1,234.00)**
	• **Y/N -** Used with bit fields. If bit field is checked, Y is reported. If field is unchecked, N is reported.
	• **YES/NO** - Used with bit fields. If bit field is checked, YES is reported. If field is unchecked, NO is reported.
	• **T/F** - Used with bit fields. If bit field is checked, T is reported. If field is unchecked, F is reported.
	• **TRUE/FALSE** - Used with bit fields. If bit field is checked, TRUE is reported. If field is unchecked, FALSE is reported.
	• **1/0** - Used with bit fields. If bit field is checked, 1 is reported. If field is unchecked, 0 is reported.

Length	This field determines the length of the column in the exported data file. This is the maximum amount of characters allowed to be reported in the column. Data which exceeds the defined length is truncated on the right side. Zero padding is added to the left of a value. Space filling is added to the right of a value. A length must be defined for each field when exporting the filter in Fixed Width format within the Data Export tool.

Field	Description
Save To	Indicates whether the filter saves to the current user, a user group(s), or a specific folder. If a filter is saved to more than one user group, a separate copy is stored for each group. Each group can independently edit the filter without affecting another group's copy. If a filter with the same name already exists within a group, the filter name is appended with a number in parentheses indicating an incremented version number (*i.e.,* HonorStudents already exists for a group, so saving a new filter with the same name appends the name to HonorStudents(2)). If the filter is saved across multiple groups, the filter name is displayed only as an appendage in groups where a filter with the same name already exists.
Test	This field allows users to test and preview a filter before saving it. Test results display in a separate window. To view the test filter, pop-up windows must be enabled on the web browser.
Save	Saves the filter.

Grouping and Aggregation Descriptions

Grouping and aggregation place results into groups, and calculations can be performed on the results. Aggregations are displayed at the bottom of each data group when extracting data. These options are not available for fixed-width output formats.



Click here to expand...

The following describes the available options.

Field	Description
Grouping	This is the order in which each group is reported. Users can report up to 5 tiers (or groups).
Group By	Determines which field is in the group and reports aggregate/sub-totals. Only fields included within the filter are available for selection.
Group Order	Determines how group aggregate/sub-totals are reported when exported via the Data Export tool.
Aggregate/Sub Total by	Determines which field within the filter is used for the 'Group by' fields. For example, a user creating a behavior Ad hoc filter who chooses to Group By behavior events and Aggregate By personID using an Aggregate Type of Distinct Count produces the number of students per Behavior Event Type.

Field	Description
Aggregate Type	Determines which calculation is applied to the group when calculating and reporting aggregate/sub-totals. For example, a group containing student last names (student.lastName) with an Aggregate/Sub Total of State ID (student.stateID) and an Aggregate Type of Distinct Count reports individual groups based on student last names with a count of how many students within that group have distinct State IDs. Aggregate Types include: • **Record Count** - Indicates the total number of records in the group. • **Distinct Count** - Indicates the total number of distinct records within a group based on the fields selected to be counted from the Aggregate By option. • **MIN—Indicates the minimum value for the designated Aggregate/Sub Total field within a group (e.g.**, an Aggregate/Sub Total for State ID (student.stateID) with a MIN Aggregate Type reports the smallest State ID value with each group). • **MAX—Indicates the maximum value for the designated Aggregate/Sub Total field within a group (e.g., an Aggregate/Sub Total for State ID (student.stateID) with a MAX Aggregate Type** reports the largest State ID value within each group). • **SUM** - Indicates the sum of all values within a group for the Aggregate/Sub Total field selected (*i.e.,* an Aggregate/Sub Total for Present Minutes (attendanceDetail.presentMinutes) with a SUM Aggregate Type reports a sum of all Present Minutes with each group). • **AVG** - Indicates the average of all values within a group for the Aggregate/Sub Total field selected (*i.e.,* an Aggregate/Sub Total for Present Minutes (attendanceDetail.presentMinutes) with AVG Aggregate Type reports the average of Present Minutes for all students within each group) See the Rules for Aggregate Calculations by Data Type table below for more information.
Save To	Indicates whether the filter saves to the current user, a user group(s), or a specific folder. If a filter is saved to more than one User Group, a separate copy is stored for each group. Each group can independently edit the filter without affecting another group's copy. If a filter with the same name already exists within a group, the filter name is appended with a number in parentheses indicating an incremented version number (*i.e.,* HonorStudents already exists for a group, so saving a new filter with the same name appends the name to HonorStudents(2)). If the filter was saved across multiple groups, the filter name is displayed only when a filter with the same name already exists in a group.

Field	Description
Test	This field allows users to test and preview a filter before saving it. Test results display in a separate window. To view the test filter, pop-up windows must be enabled on the web browser.
Save	Saves the filter within Infinite Campus. The filter is now available for use in all Ad hoc Filter fields throughout Infinite Campus (if the user is part of the user group the filter was saved to).

Rules for Aggregate Calculations by Data Type

The following table describes all rules for allowing or disallowing aggregate calculations based on data type:

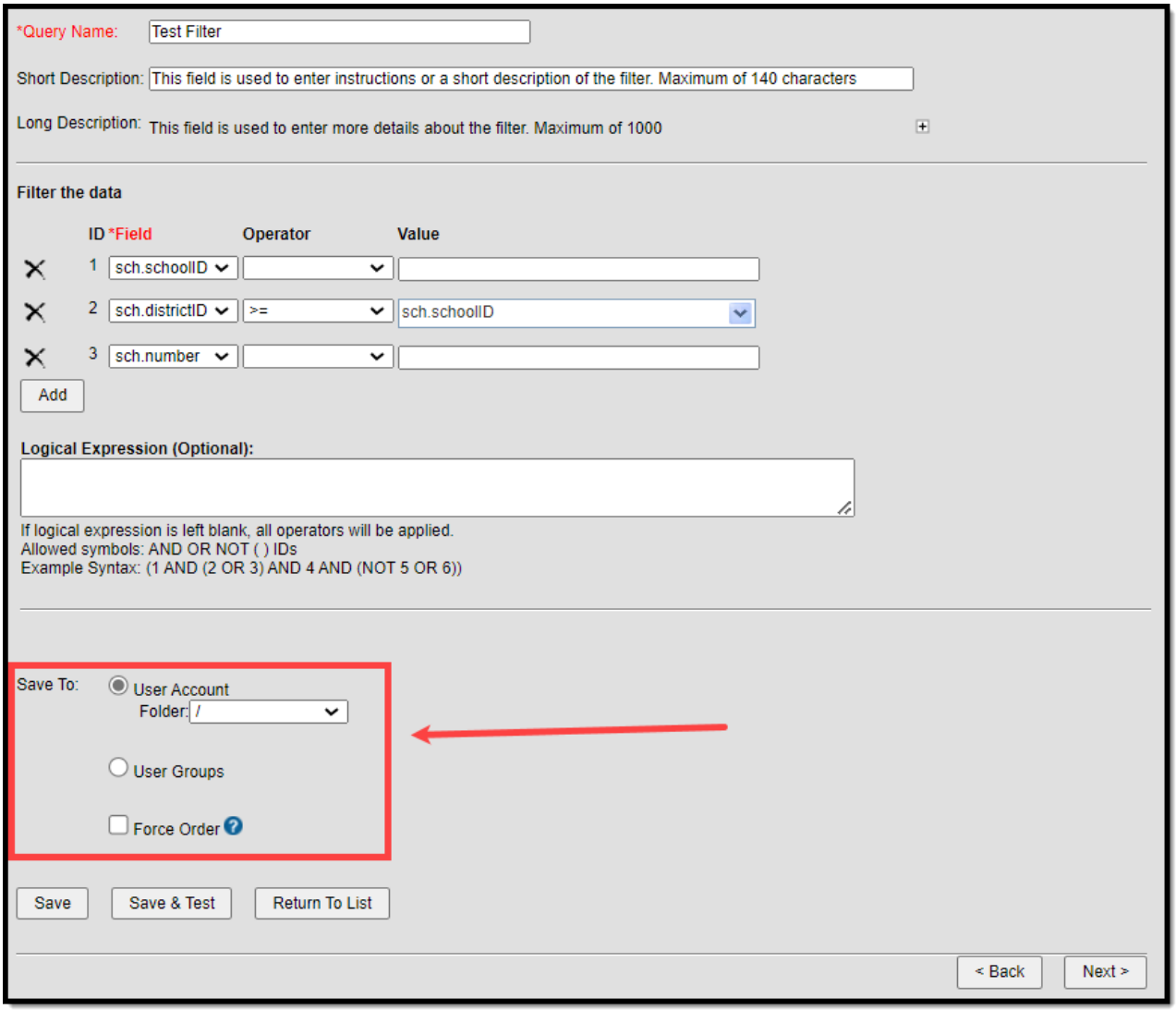
Data Type	Number	Float	String	Date	Text	Bit
MIN	Yes	Yes	Yes	Yes	Yes	Yes
MAX	Yes	Yes	Yes	Yes	Yes	Yes
AVG	Yes	Yes	No	No	No	No
SUM	Yes	Yes	No	No	No	No
Record Count	Yes	Yes	Yes	Yes	Yes	Yes
Distinct Count	Yes	Yes	Yes	Yes	Yes	Yes

Manage Filters

Save Filters to Folders

Ad hoc filters can be saved to specific folders created within the Filter Designer tool, User Accounts, or User Groups.

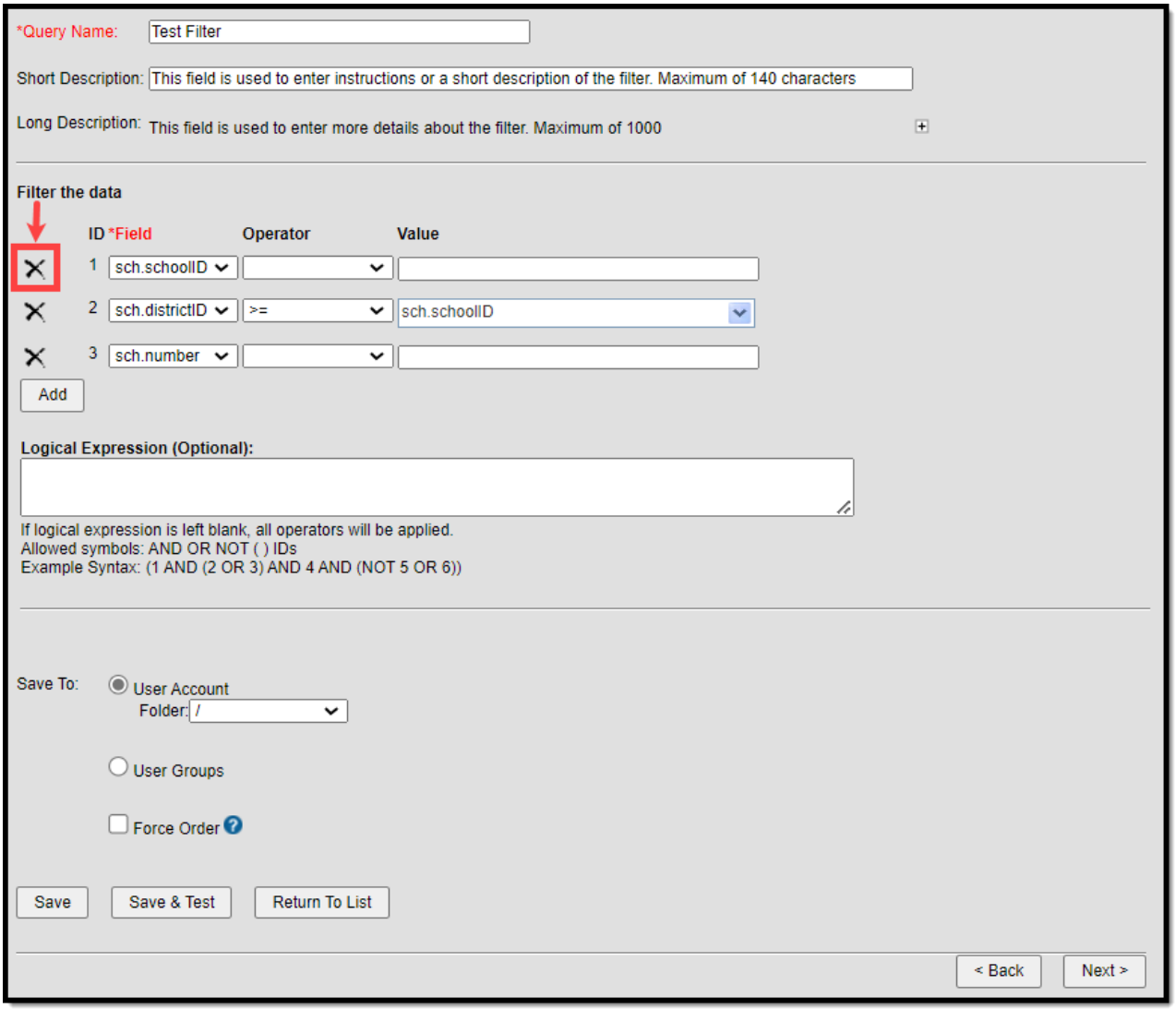
For complicated filters that report data from several calendars and/or have many fields from many different areas, mark the **Force Order** checkbox. When marked, the database fields in the query are executed in a particular order to increase the filter's performance. When a filter takes several minutes to generate, try generating it again with this checkbox marked. Marking this on every filter is not recommended.



Remove Fields from the Filter Parameters Editor

Fields can be removed from the Filter Parameters editor without being removed from the filter as a whole. This allows users to reduce the Filter Parameters editor to only those fields in which operators are assigned or only those fields in which the user wants to see.

Fields removed from the Filter Parameters editor are not removed from the filter; they are only the user's view of the editor.



Select the X next to each field to remove fields from the Filter Parameters Editor.

Removing a field from the list does not remove it from the filter output.

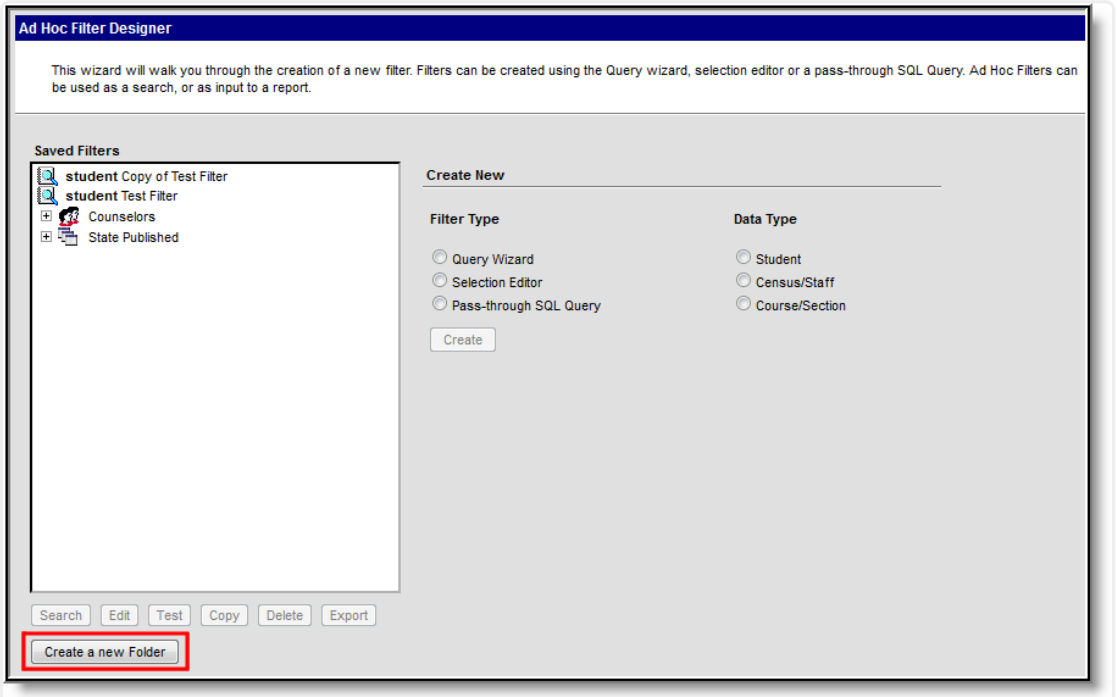
All fields not assigned an Operator were removed, and the field IDs were automatically renumbered. The Logical Expression automatically updates to match new field IDs.

The Filter Designer tool allows users to create folders for organizing and storing Ad hoc filters. Folders can be organized in a hierarchy format, where sub-folders exist within parent folders. By creating folders, users can better manage large volumes of existing Ad hoc filters and group them in a logical order.

If a field in the query has been deactivated (displays in red), use the Element Replacement Tool to update the filter. This removes the deactivated field and adds the equivalent field to the filter.

Create Folders for Filters

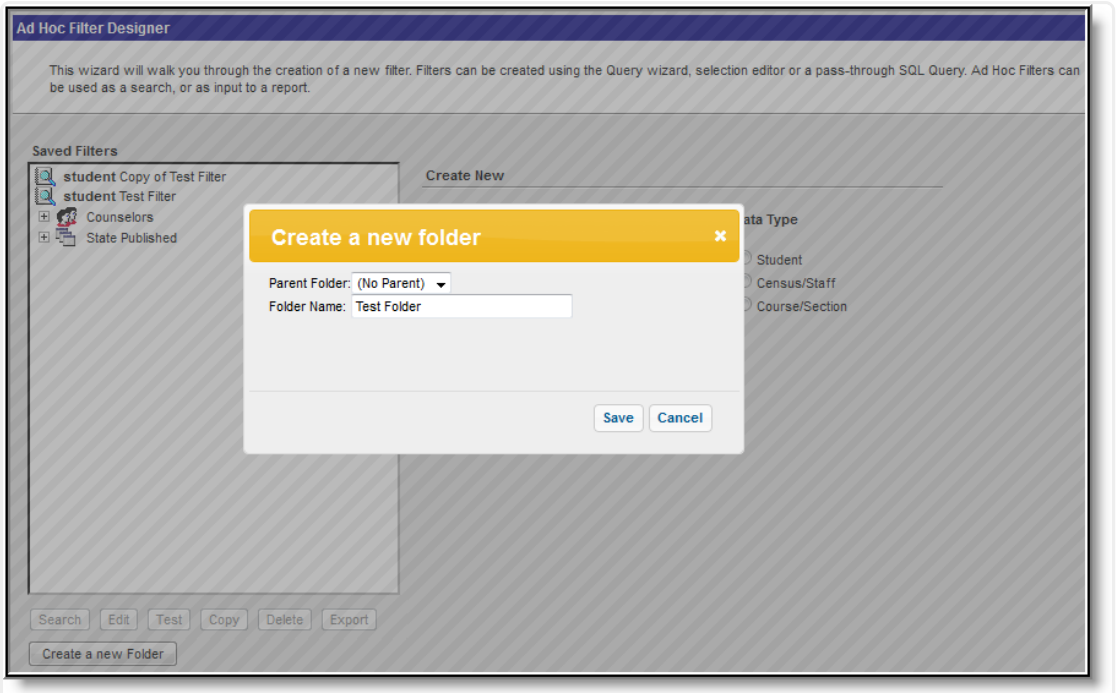
Folders allow users to better manage Ad hoc filters within the Filter Designer tool.



Create a New Folder

Click here to expand...

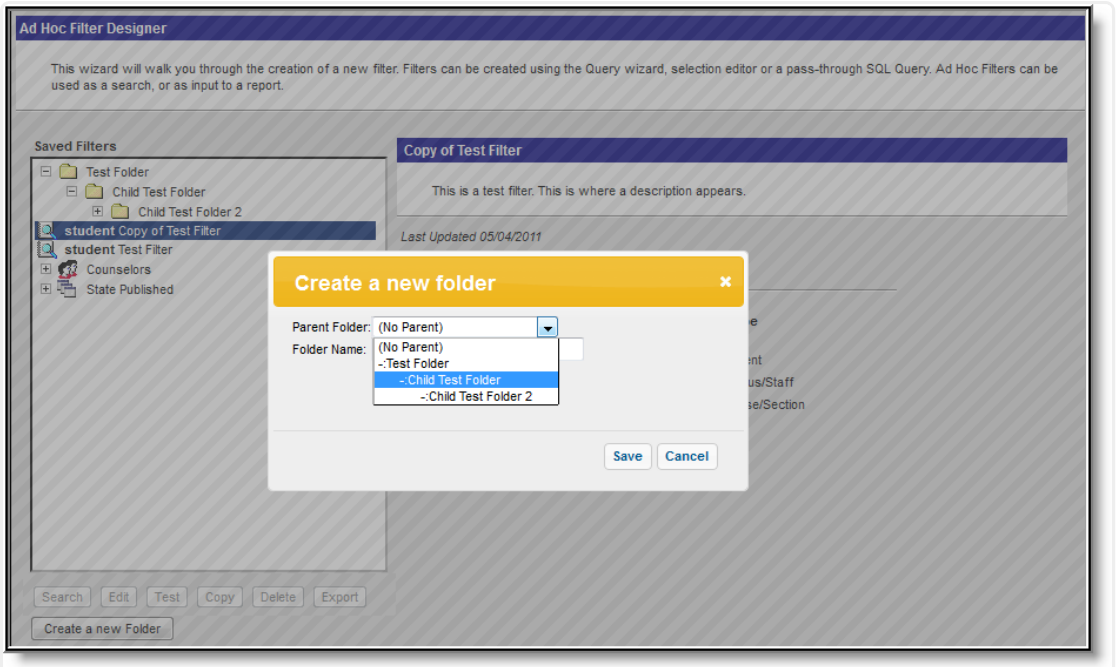
To create a new folder, select the **Create a new Folder** button. The **Create a new folder** editor displays.



Create a New Folder Editor

If the folder should not be tied to a parent folder, leave the **Parent Folder** field as (No Parent), enter a **Folder Name** and select the **Save** button. The folder displays in the **Saved Filters** field and is now available for storing Ad hoc filters.

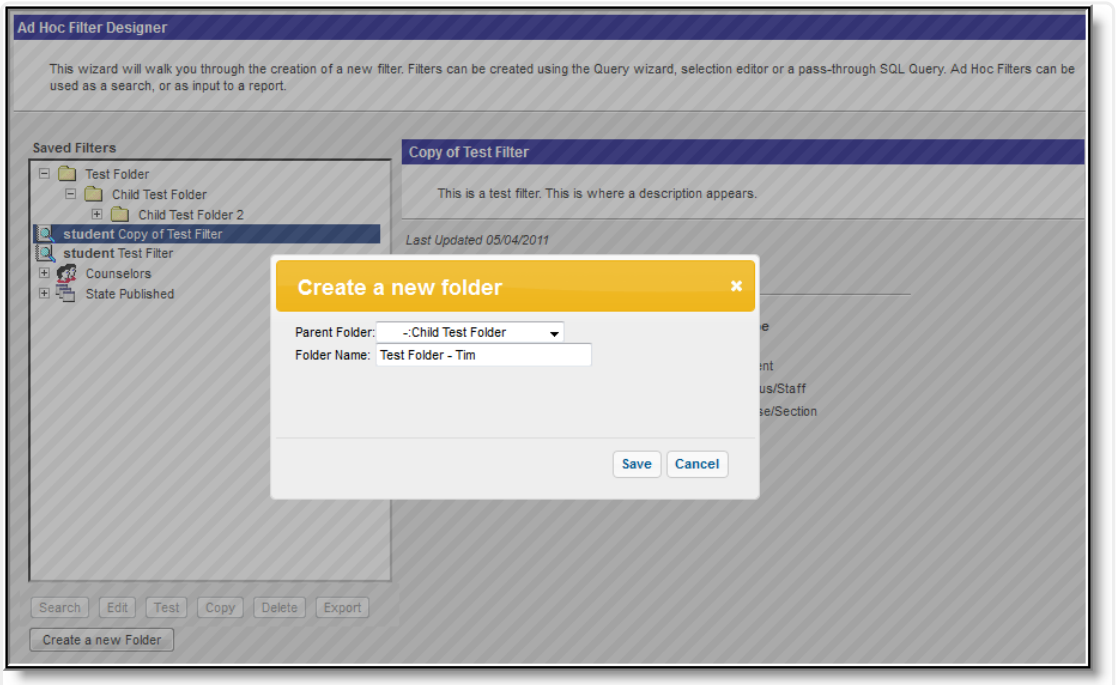
If the folder should be assigned to a parent folder, select the parent folder from the **Parent Folder** field.



Selecting a Parent Folder

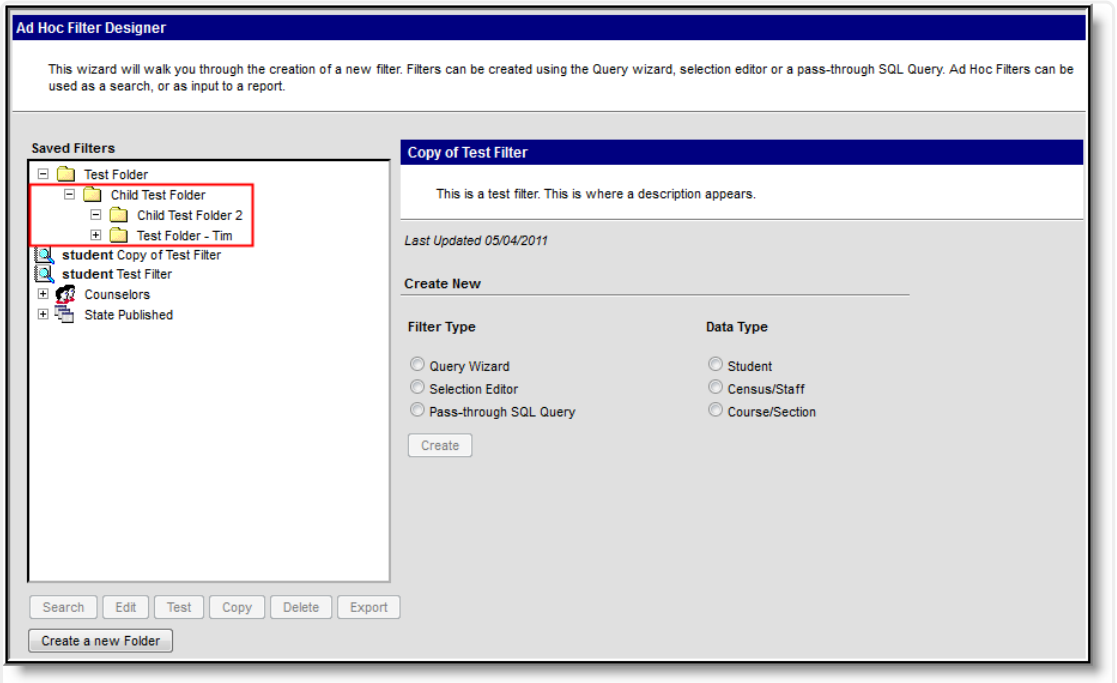
Locate the appropriate parent folder. The indentation next to each folder name indicates its

relationship to the previous folder (*i.e.*, the Grandchild 1 folder is indented two times because it exists within the Child 1 (Testing) folder which exists within the Parent Folder (Testing) folder). In the example above, the folder being created exists within the Child Test Folder parent folder.



Entering a Folder Name

Once the parent folder has been selected, it displays in the **Parent Folder** field. Enter the **Folder Name** of the folder being created and select the **Save** button. The folder displays in the Saved Filters field and is now available to store Ad hoc filters.

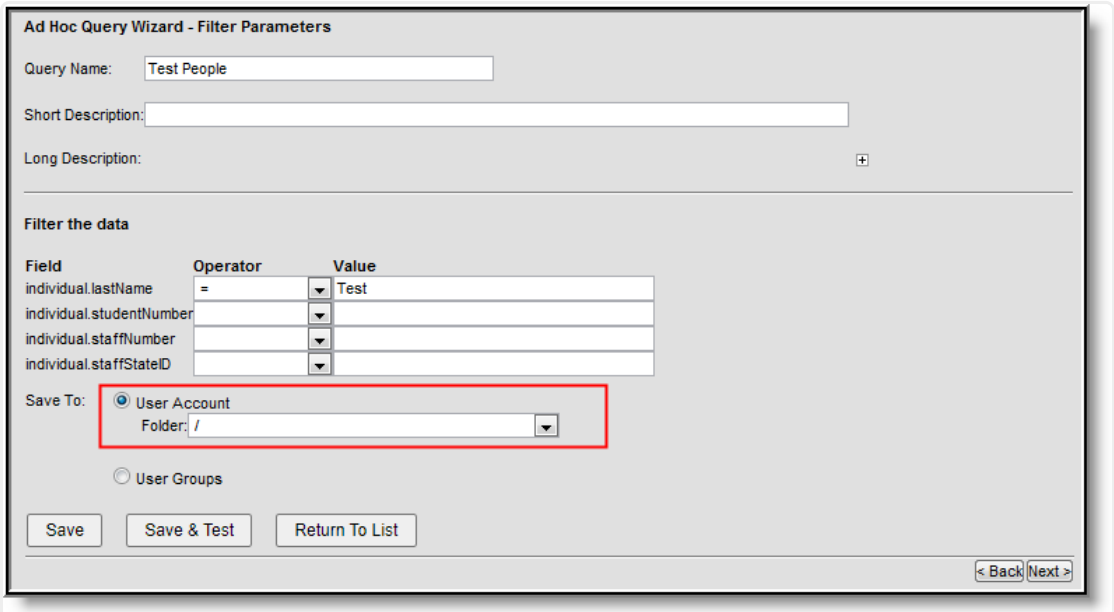


Created Folder

As the example above shows, the created folder Test Folder - Tim now exists within its parent folder Child Test Folder.

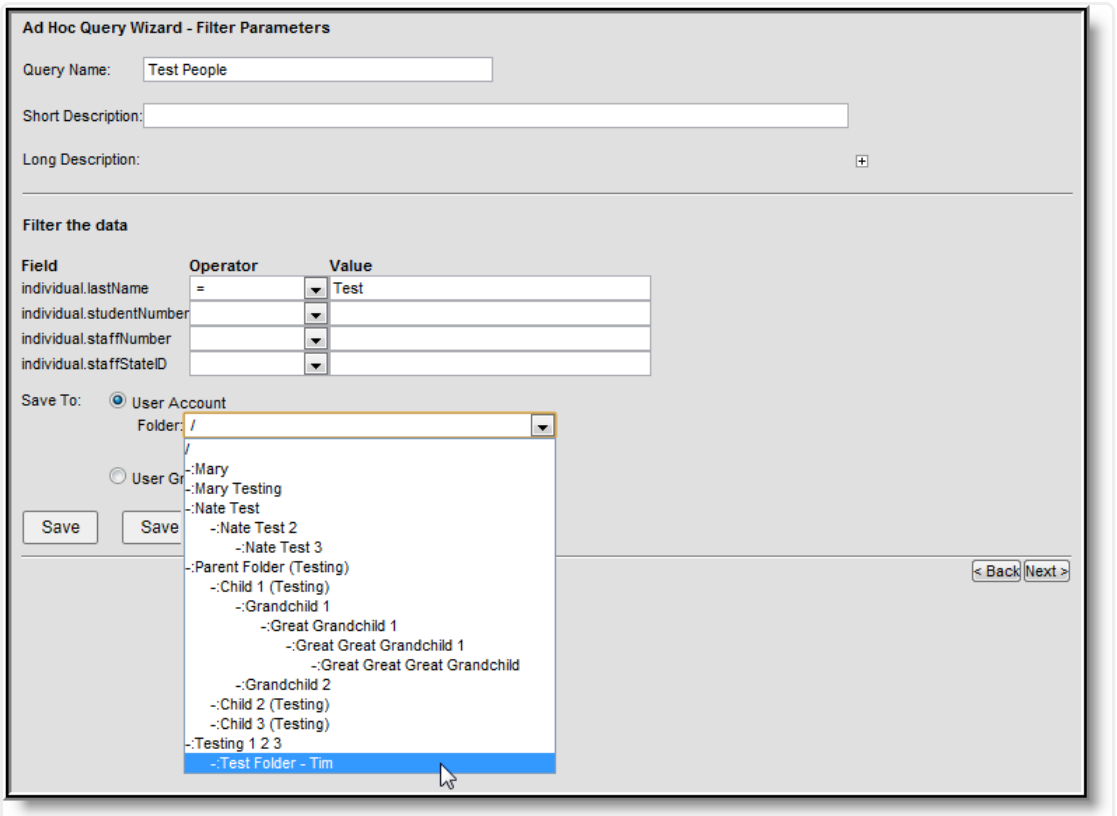
Add a Saved Query to a Folder

Once folders have been created, Ad hoc filters can now be assigned to those folders.



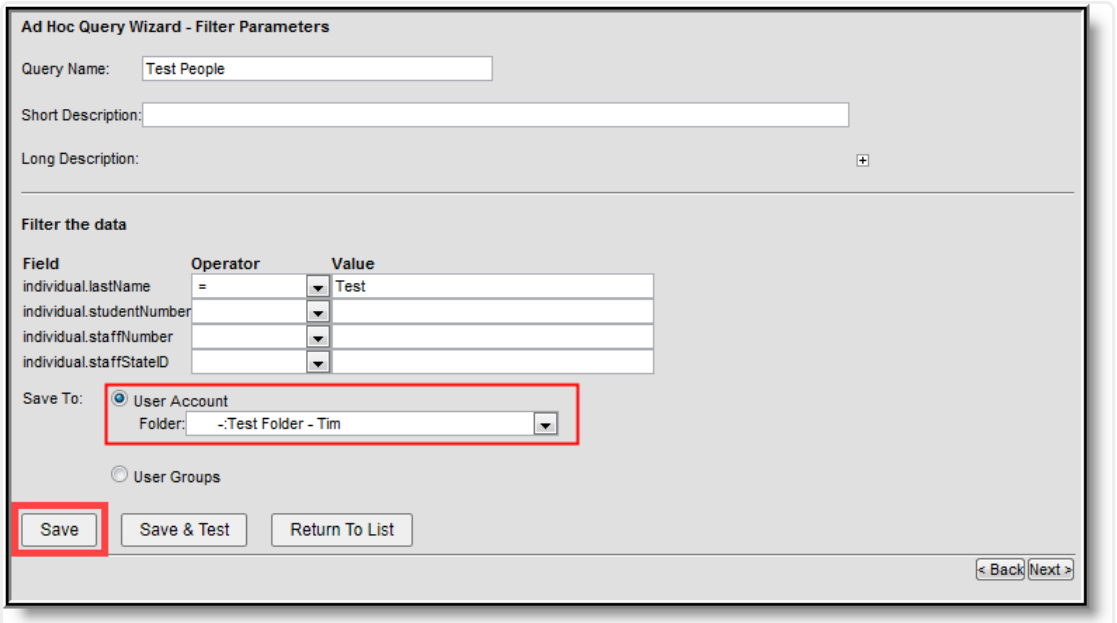
Saving an Ad hoc Filter to a Folder

To assign an Ad hoc filter to a folder, click the **User Account** radio button and select the folder from the **Folder** field.



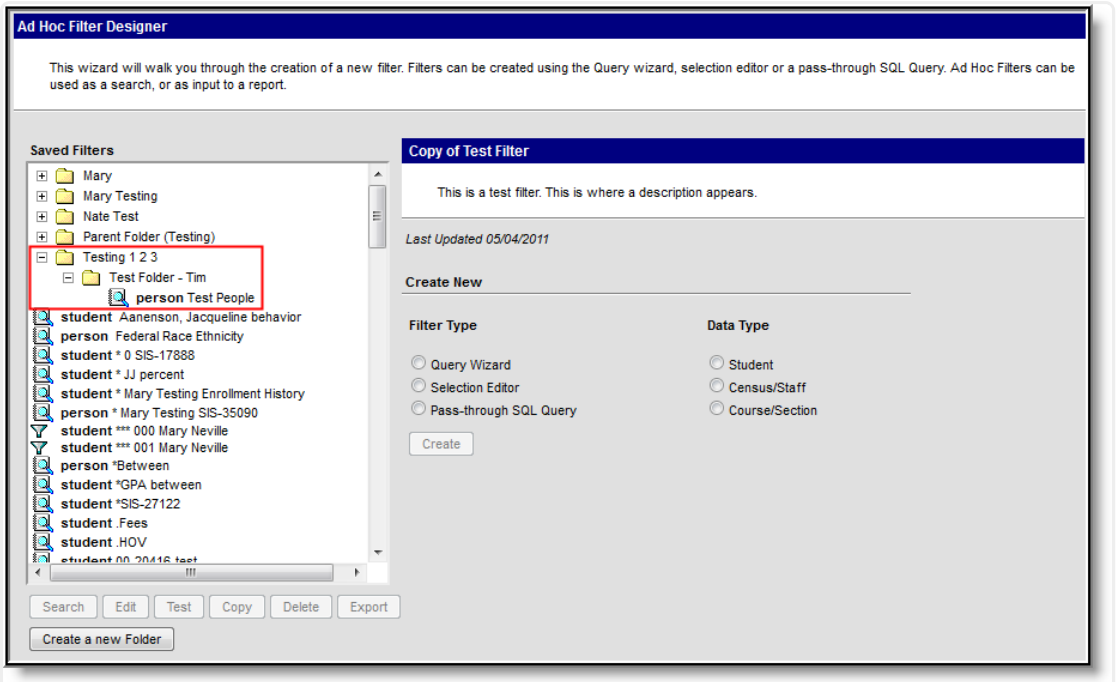
Selecting the Saved Folder

In the example above, the Ad hoc filter is being assigned to the Test Folder - Tim folder.



Saving the Ad hoc Filter to a Folder

Once a folder is selected, the **Folder** field displays its name. Select the **Save** button to save the filter to the folder.

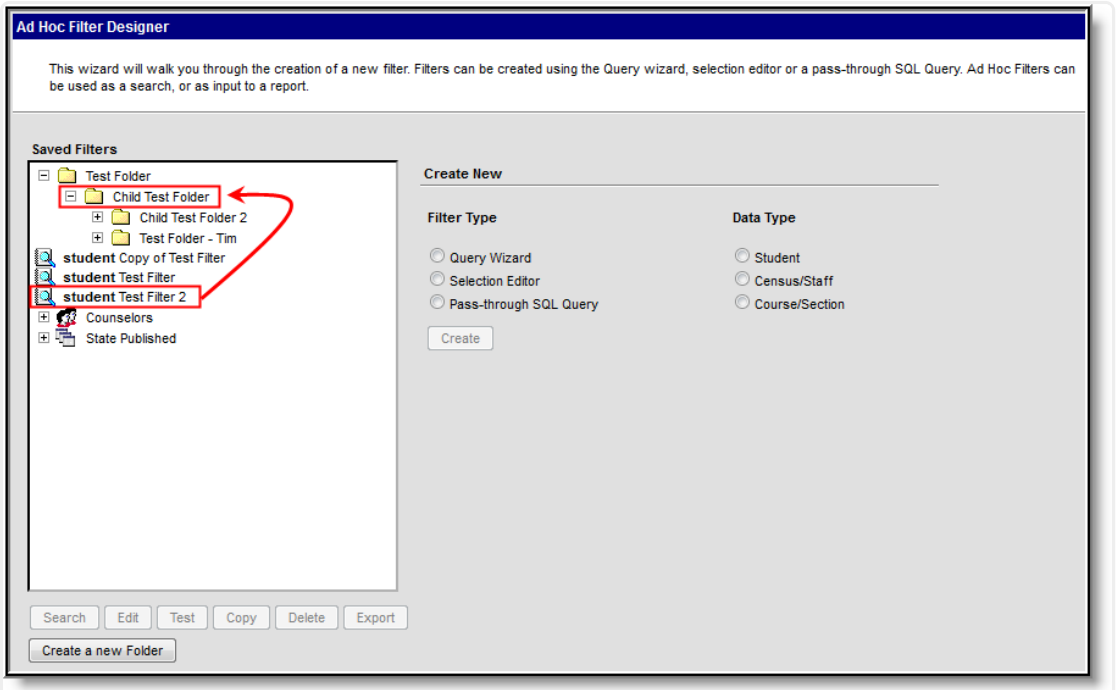


Viewing the Saved Filter in the Folder

The Ad hoc filter is now saved and accessible within the assigned folder.

Move Filters between Folders

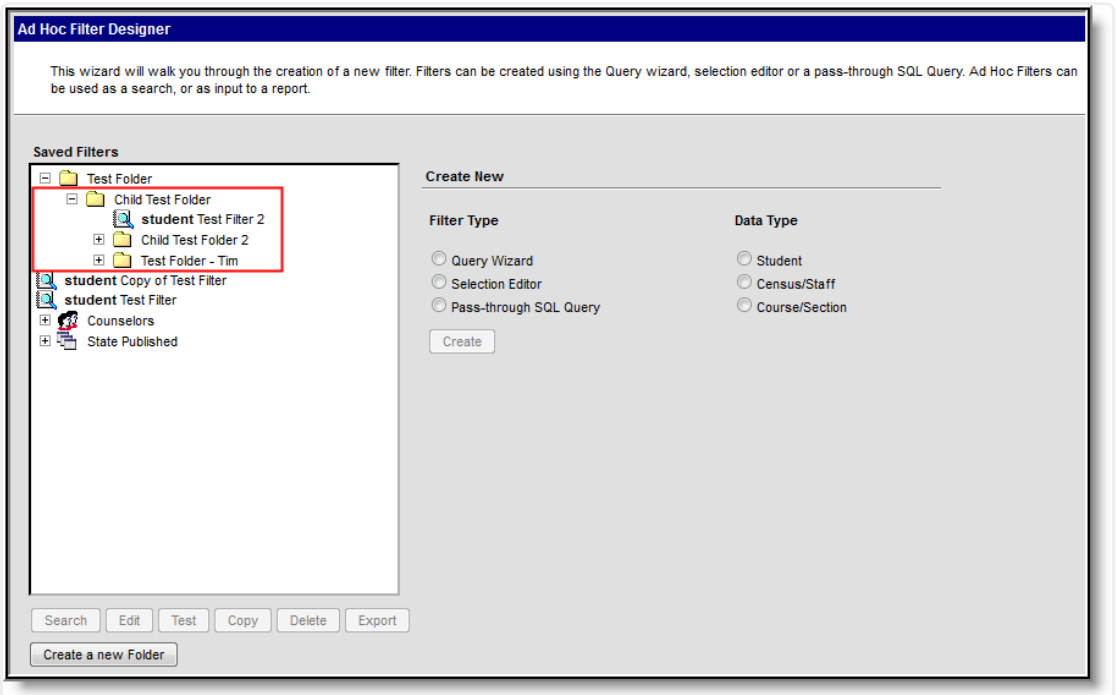
Ad hoc filters can be easily moved and organized between folders.



Moving an Ad hoc Filter to a Folder

To move an Ad hoc filter into an existing folder, left-click, hold, and drag the filter into the

designated folder. A pop-up message displays, asking the user to confirm the action. Select the **OK** button to move the Ad hoc filter.



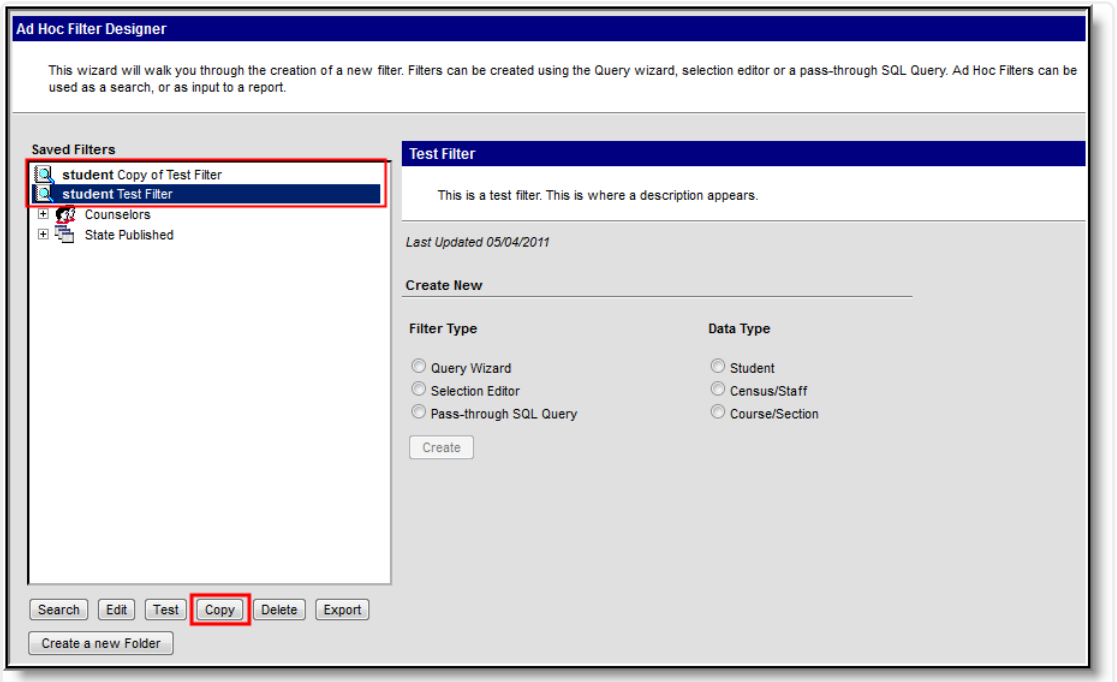
Viewing a Moved Ad hoc Filter

The moved filter now displays under the appropriate folder. This functionality moves filters in, out, and to another folder.

Existing filters can be easily copied if desired. This maintains the original version of the filter and lets users change a filter to add new fields and functions.

Copy Filters

Filters can be copied for additional editing. Select a saved filter and click the **Copy** button. A pop-up message displays indicating the filter has been copied. Copied filters are named Copy of [Original Filter Name].

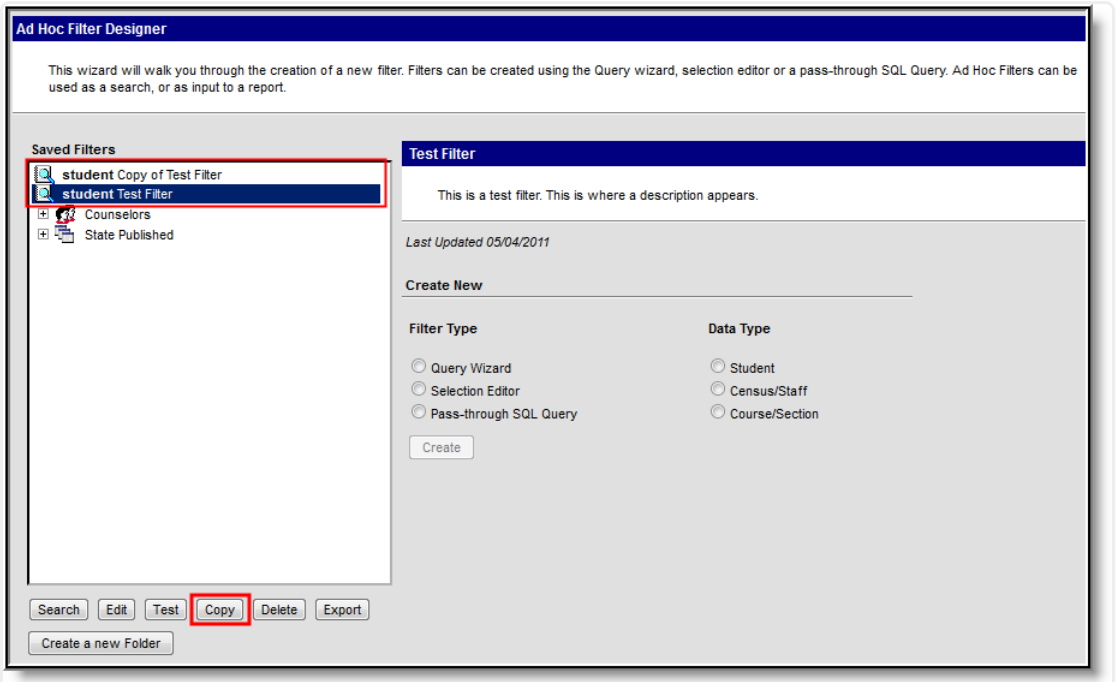


Copied Filters

Delete Filters

A saved filter created by a user can also be deleted by that user. However, because filters can be shared with other users, only the person who created the filter can delete it.

District users cannot delete State-Published filters.



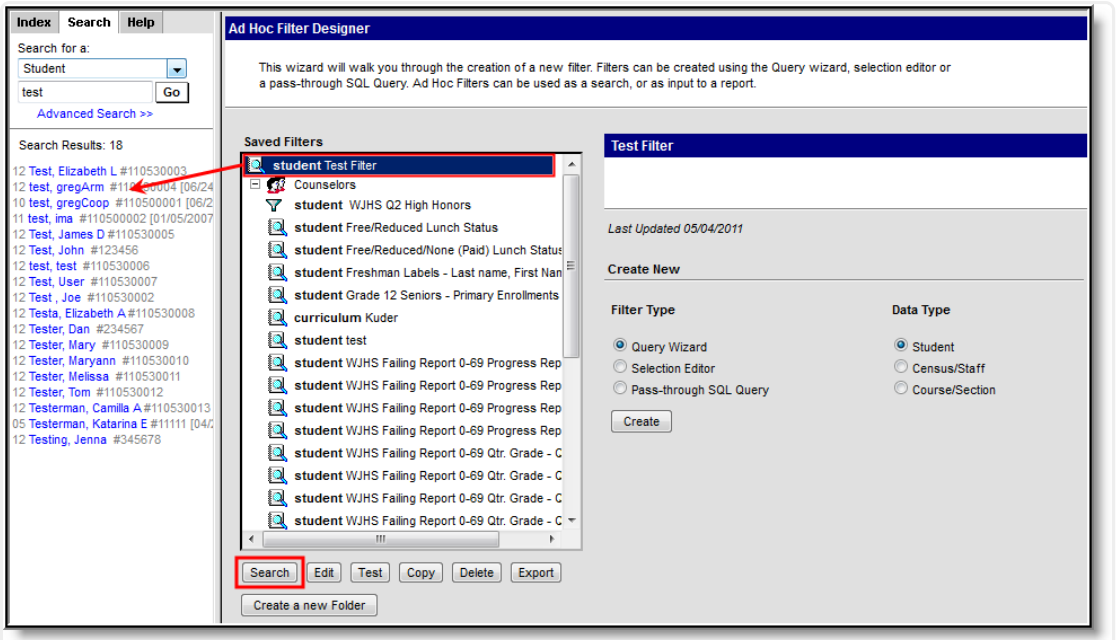
Deleting a Filter

Select a filter from the Saved Filters window and click the Delete button to delete it. A pop-up message confirms the deletion. You can also delete multiple filters by holding the **Ctrl** key, selecting each filter, and clicking the **Delete** button.

Modify a Query Created by Another User

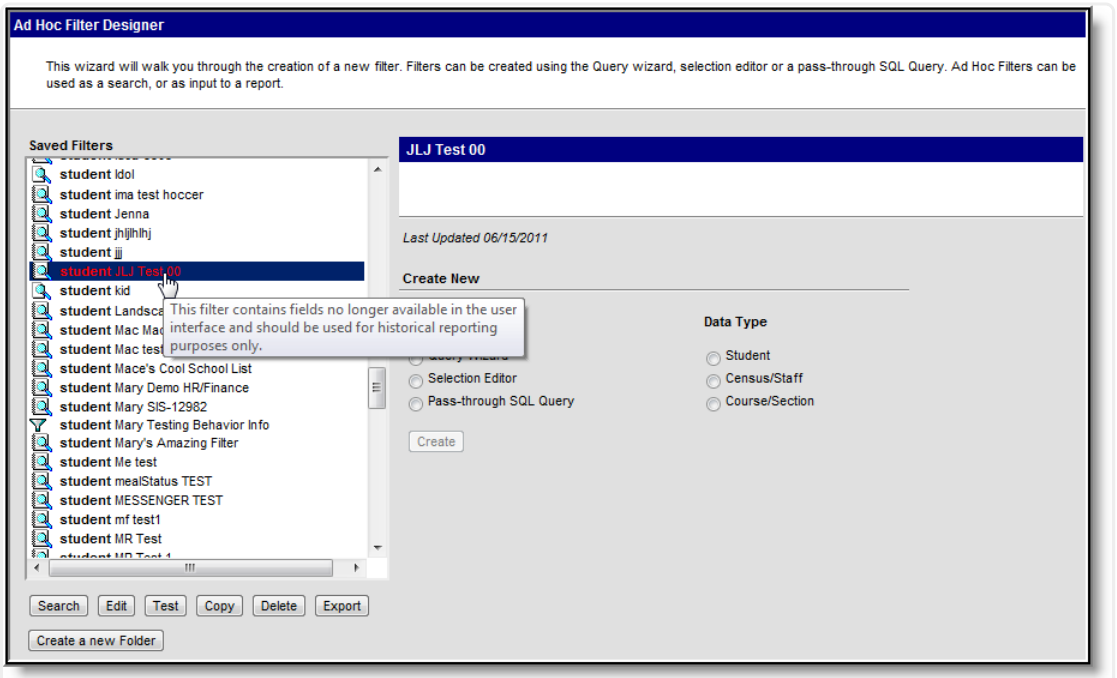
Saved filters can be edited anytime by selecting the filter and clicking the **Edit** button. This displays the filter so users can modify the selected fields and verify the operations and export options.

Search results on the Search tab can be populated with saved filters. When a saved filter is selected, click the **Search** button. Results returned in the filter displays in the Search tab.



Searching with a Filter

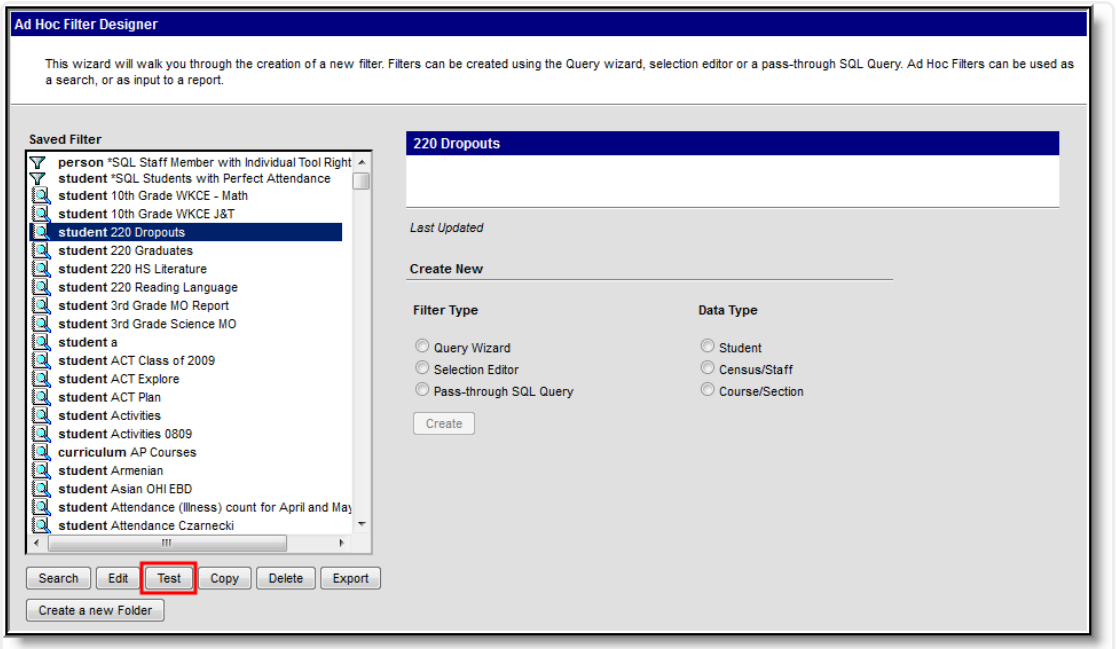
If a saved filter contains deprecated fields, the filter is highlighted in red within the Saved Filters window.



Filter Containing Deprecated Fields

Test Saved Filters

Select the filter from the Saved Filter window to test an existing filter and click the **Test** button. A separate window displays, displaying filter results in HTML format.

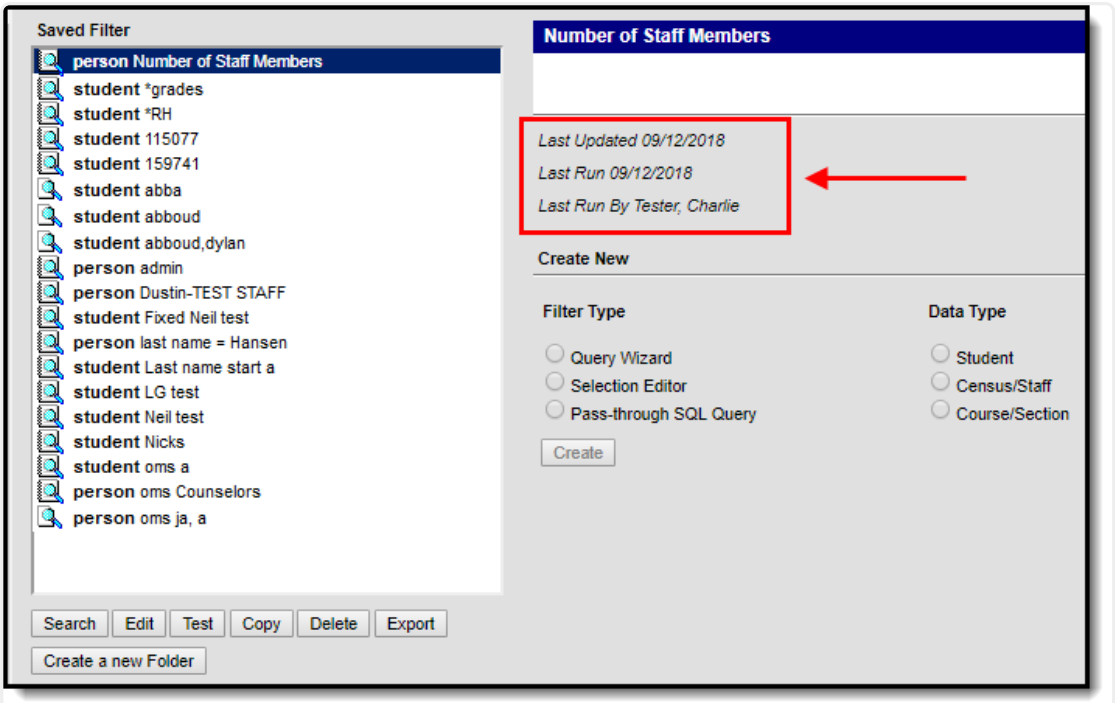


Testing an Existing Filter

Last Updated, Last Run, and Last Run By Information

Users can view the last time an existing filter was updated, the last time a test of the filter was run, and who ran the last test of the filter.

> If the timestamp or user is unknown, a value of Unknown is reported.



Example of a Filter Tracking Information