

AccuTrack / AccuSQL Tech Tips: How do I Use Query Generator?

As anyone who uses AccuTrack / AccuSQL knows, there are many reports and report filters built into the program. While numerous reports are indeed available, you may find times where a built-in report does not provide the information you really need. In the past, you were required to either use the challenging “Customize Report” feature to build the report you wanted, or would have to request that we build it for you for a future update to the software.

I think we are all glad that those days are pretty much over since we built the Custom Query Generator into the program. Query Generator is a very powerful tool in AccuTrack / AccuSQL used to build your own custom queries and then decide how you want the output to be displayed.

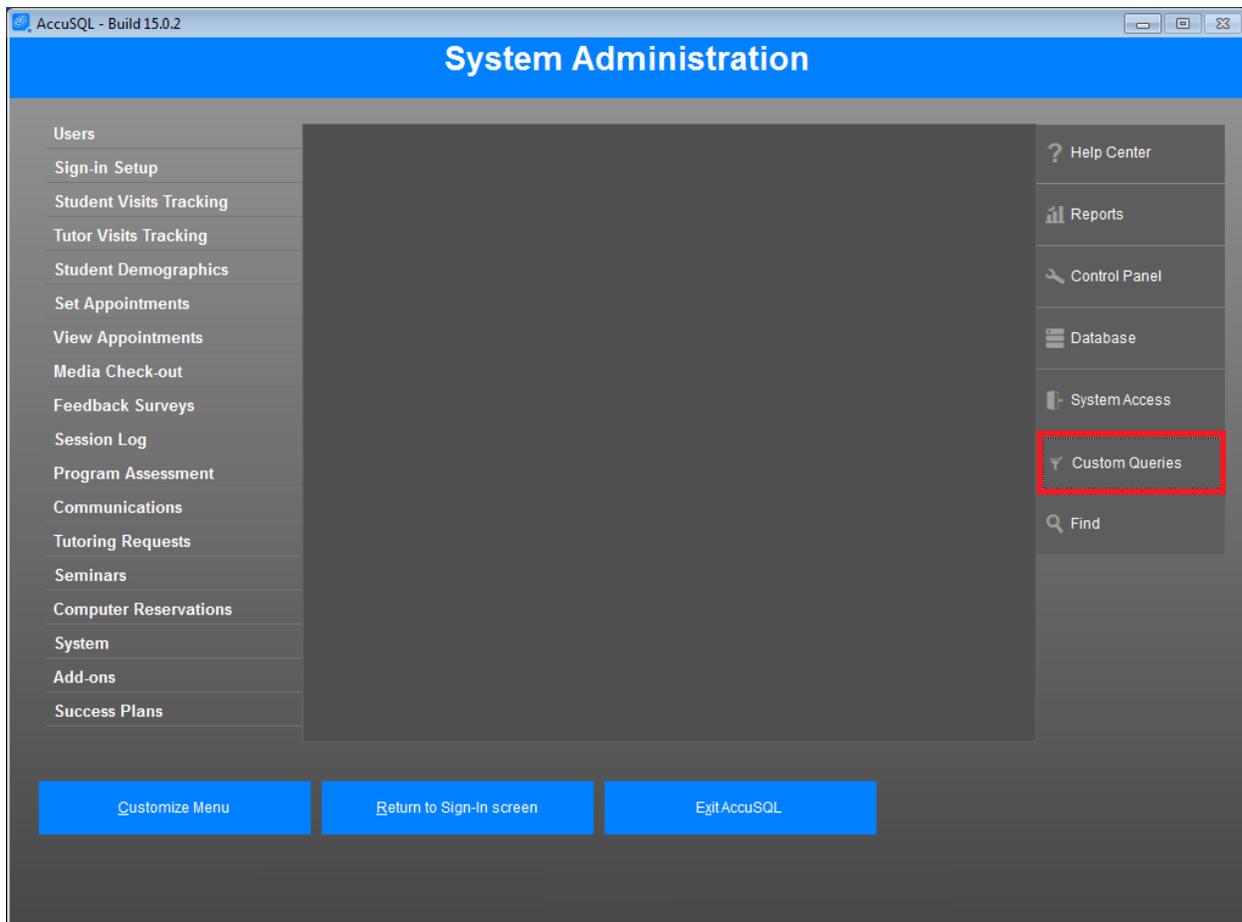
In this article, I will be building a custom query to show you how it works. After you read this article, I would encourage you to try to build some of your own custom queries.

In this example, I will be building a query that will show all of the sign-ins and sign-outs for my students between June 1, 2015 and June 1, 2016. In addition, I want to collect the following information:

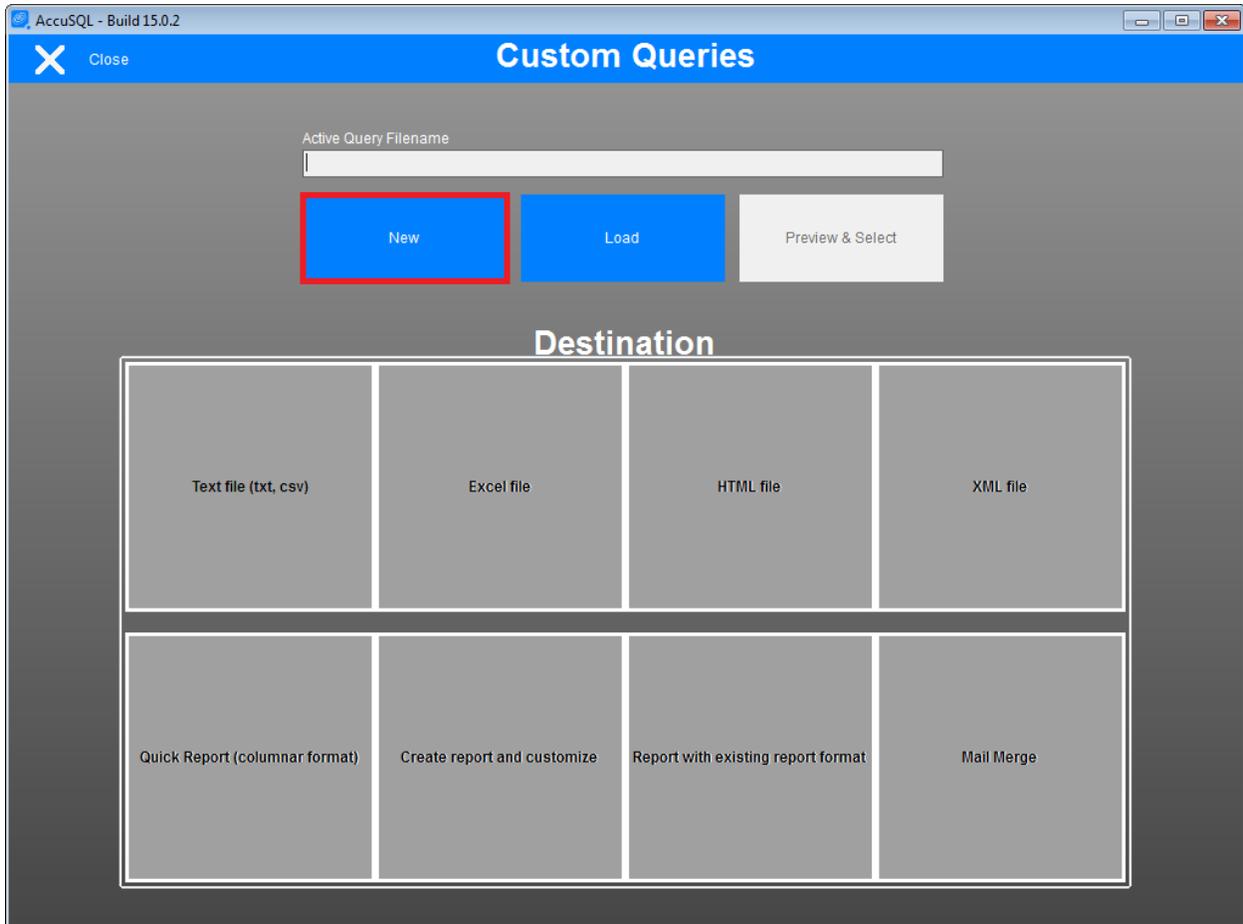
**Student ID, Student First Name, Student Last Name, and Student Group Name
Category, Activity ID and Activity name (signed into)
Tutor First Name and Tutor Last Name (seen)**

Further, I only want to see these sign-ins and sign-outs for students that are in an At Risk student group that I previously created (you can review the AccuTrack / AccuSQL User’s Manual to learn more about building and using student groups).

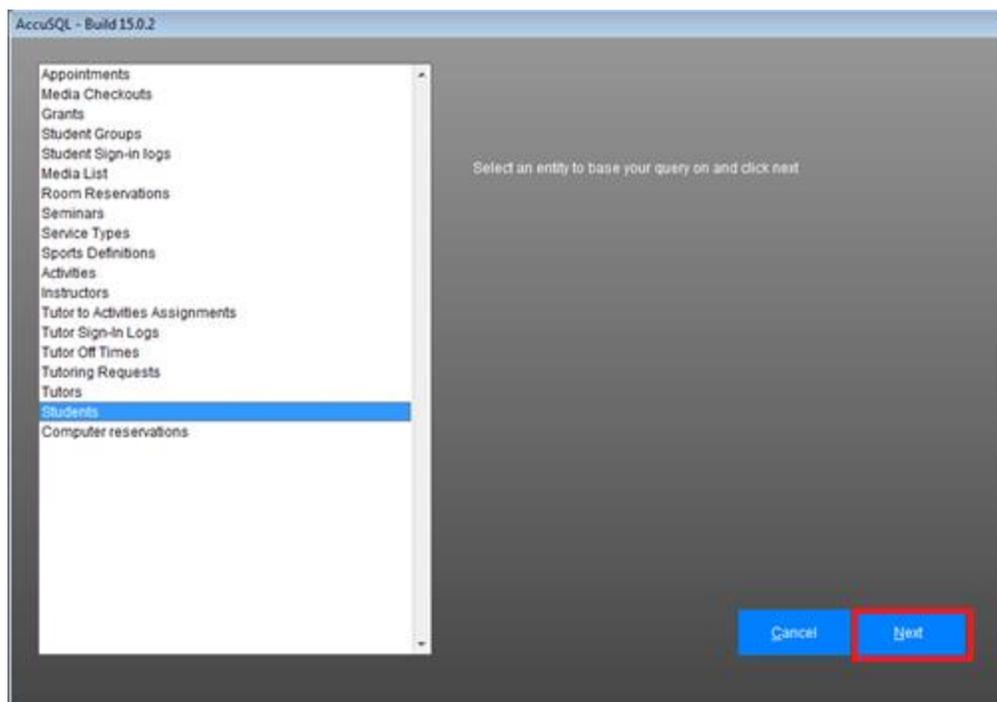
To begin the process, I select **Query Generator** in the main AccuTrack / AccuSQL menu:



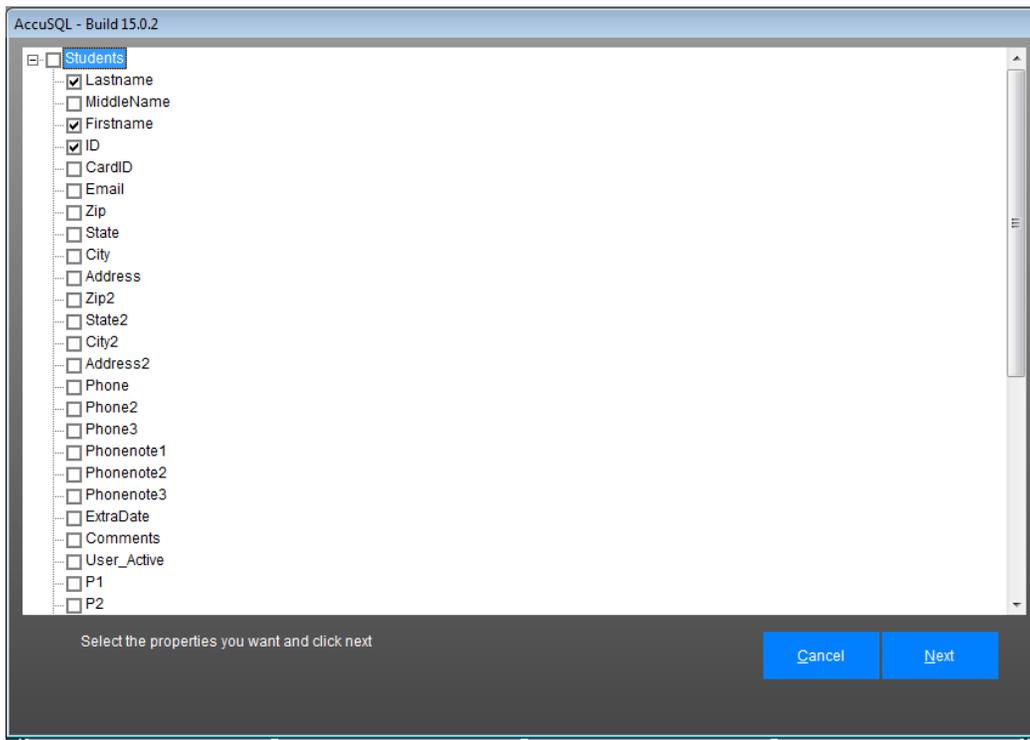
After I open Query Generator, the Custom Queries screen opens. In this screen, I can either load a query I have already built and saved, or I can create a new query. In this case, I will select **New**.



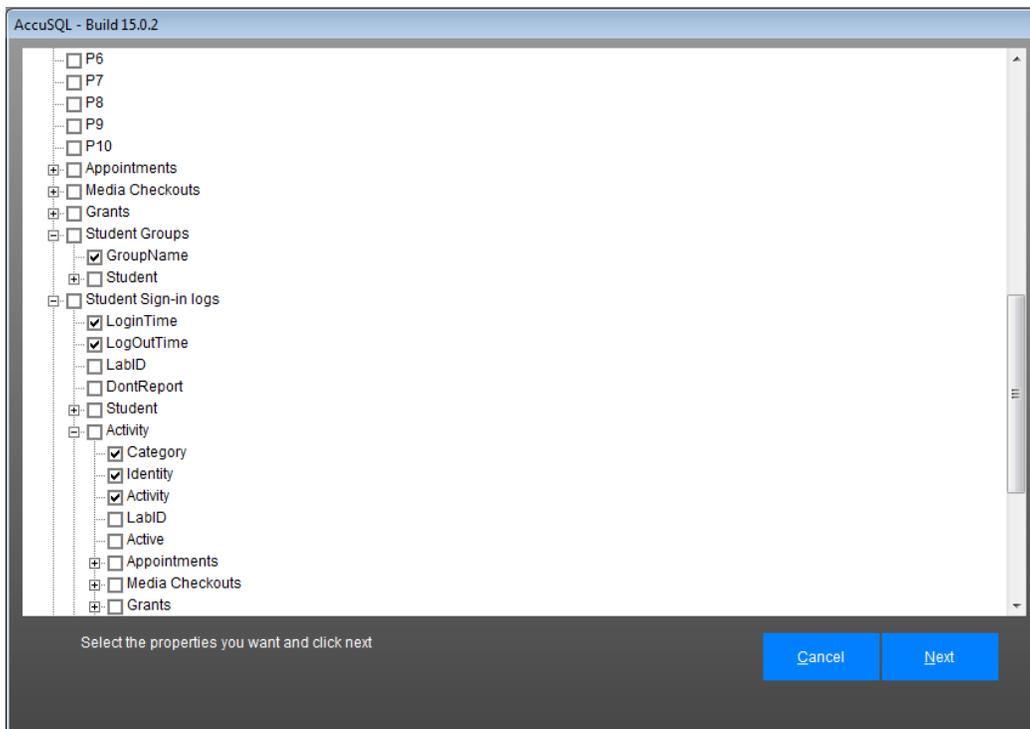
After pressing the New button, I need to select the base table for the query. Since I want to collect Student information about sign-ins and sign-outs, I will base the Query on **Students**. I could also use Tutors, or Activities as my base table in this case. Don't worry too much about the base table; it just needs to contain some of the information you are trying to collect.

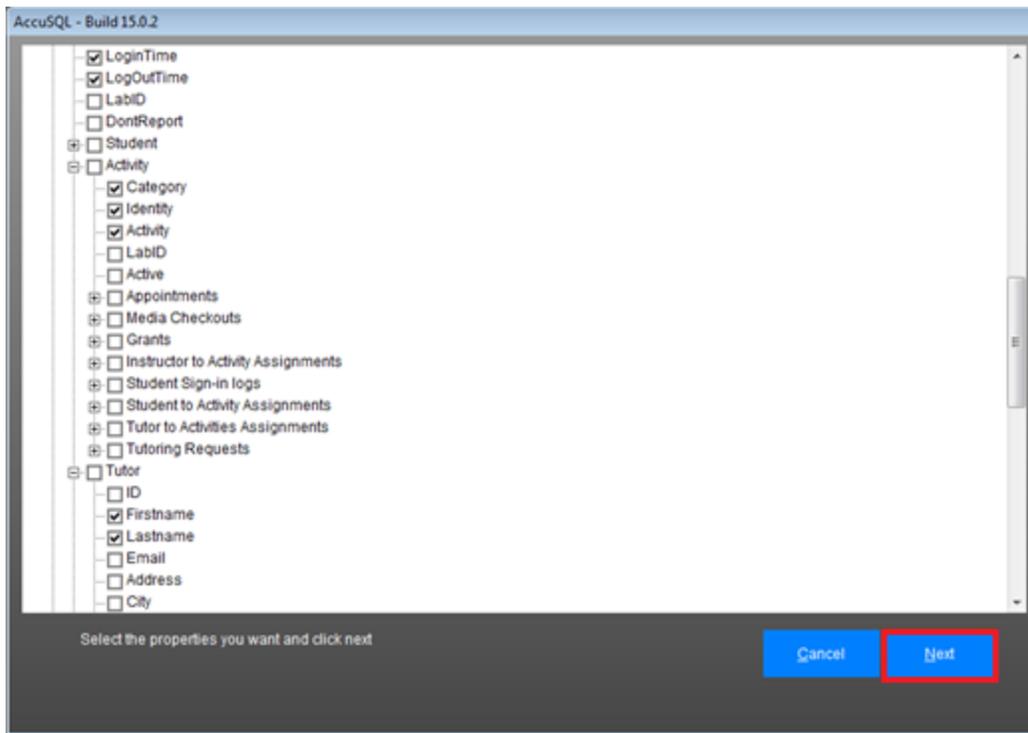


After I have selected the base table, I click **Next** and the Define Query screen opens. In this screen, I click the + sign-in the tree elements to find the data I want to add, and then click the checkboxes next to the data elements to select them. You will find many of those elements repeat themselves in various nodes due to the table associations behind the scenes. I want to select the elements typically under my top level (base) node element. I will start by selecting from **Students** table the **Lastname**, **Firstname**, and **ID** elements.



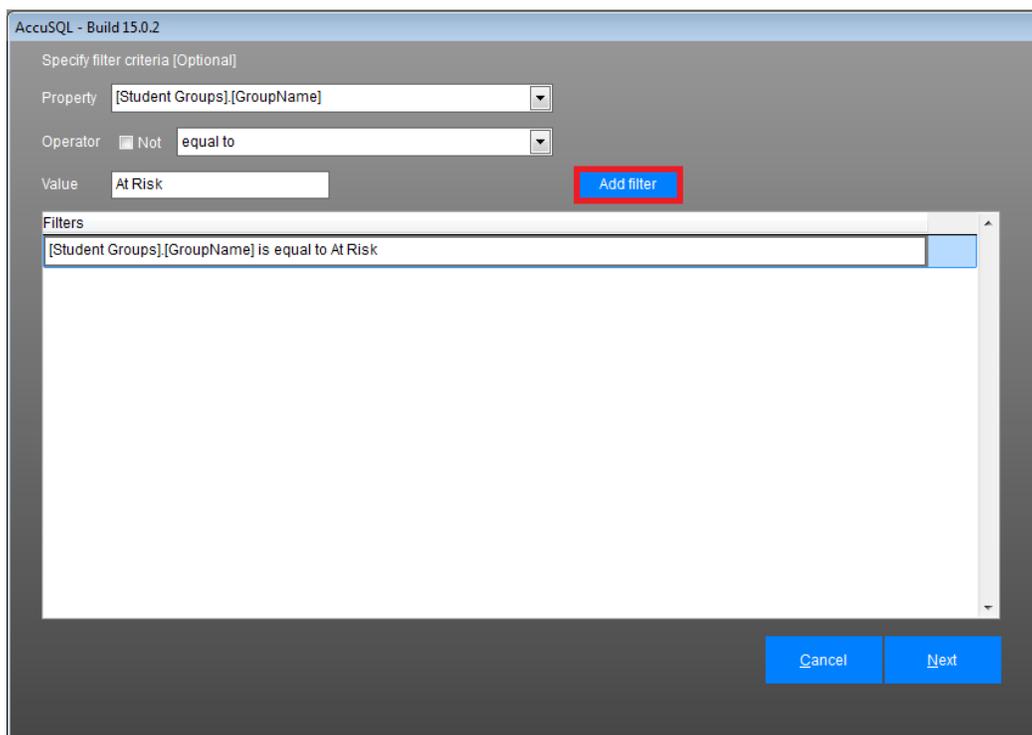
Now I need to scroll down to find the remaining elements I want to select. This would include from the **Student Groups** table the **GroupName**, from the **Student Sign-in Logs** table the **LoginTime** and **LogoutTime**, from the **Student Sign-in Logs > Activity** table the **Category**, **Identity** (activity ID) and **Activity** (name), and from the **Student Sign-in Logs > Tutor** table the **Firstname** and **Lastname**.





After I have finished selecting the elements I want to include in this query from all the tables, I click **Next**. Now it loads the screen where I can set the filtering criteria. I need to set two criteria in this example. I want to set my Student Group Name to “At Risk” and use my Login Times to only show Sign-ins between June 1, 2015 and June 1, 2016.

For the Student Group Name, I select **[Student Groups].[GroupName]** from the Property drop-down menu. Now I want to set it to Operator to the **equal to** in the drop-down box and enter the value of **At Risk**. *Note: When entering information in the value fields they are case sensitive.*



After I have defined my At Risk Student Group filter, I click **Add filter** to add it to the query.

Now I will add the filter for my date range. Remember, I want students that have signed in between June 1, 2015 and June 1, 2016. I do that by selecting the Property labeled as **[Student Sign-in logs].[LoginTime]** using the Operator of **between**. In the value field, I enter the dates as follows: **06/01/2015** and **06/01/2016**. Note: If you wish to add a time filter as well as the date, it would use this format: **06/01/2015 08:00AM** or **06/01/2016 10:00PM** for example. Now I click **Add filter** to add this item to the Filter list:

The screenshot shows the 'AccuSQL - Build 15.0.2' application window. The main area is titled 'Specify filter criteria [Optional]'. It contains the following fields and controls:

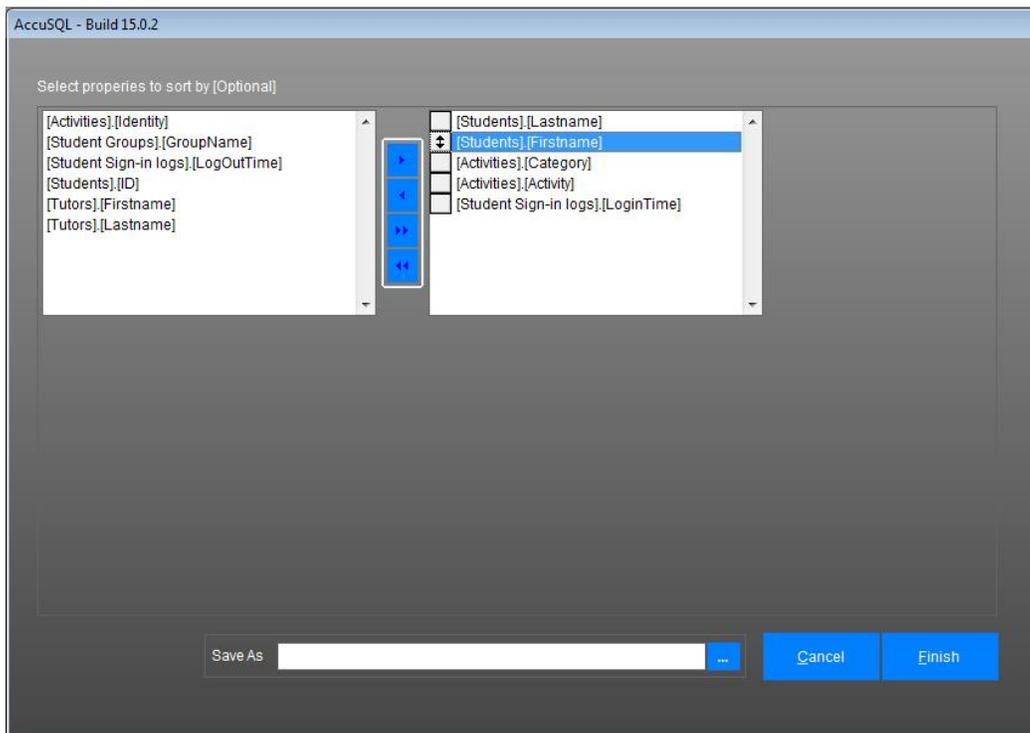
- Property:** A dropdown menu with the selected value '[Student Sign-in logs].[LoginTime]'.
- Operator:** A dropdown menu with the selected value 'between'. There is a 'Not' checkbox to the left of the dropdown.
- Value:** Two text input fields containing '06/01/2015' and '06/01/2016'.
- Add filter:** A blue button to the right of the value fields.

Below these fields is a 'Filters' list box containing two entries:

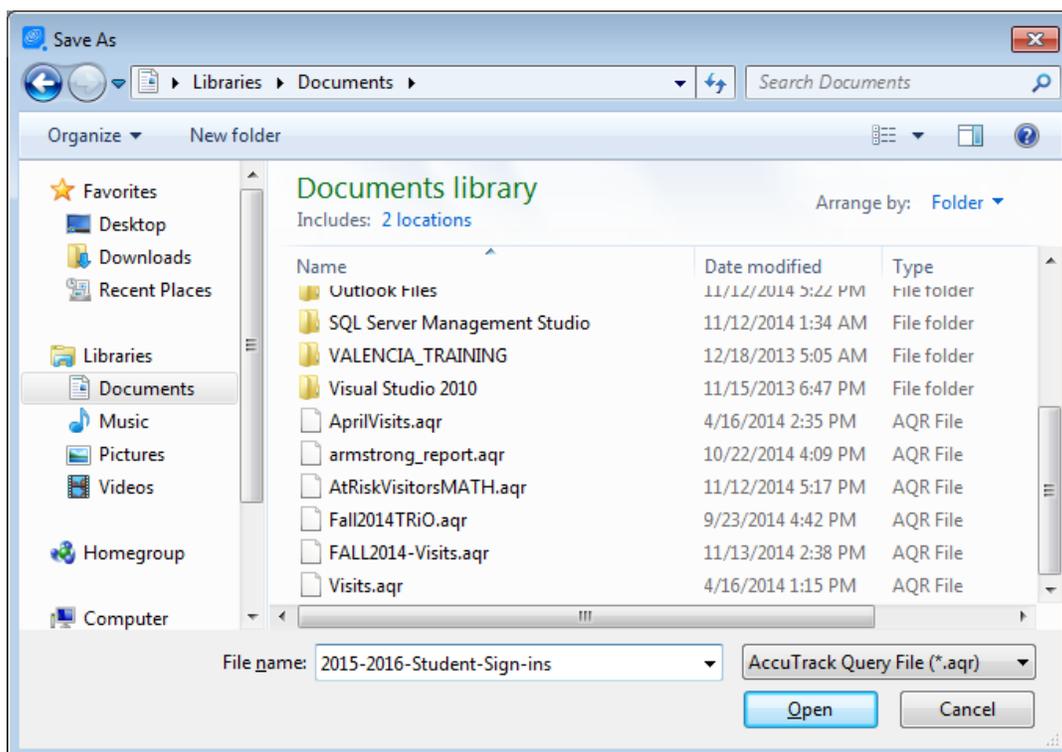
- [Student Groups].[GroupName] is equal to At Risk
- [Student Sign-in logs].[LoginTime] is between 06/01/2015 and 06/01/2016

At the bottom right of the window are two buttons: 'Cancel' and 'Next'. The 'Next' button is highlighted with a red rectangular border.

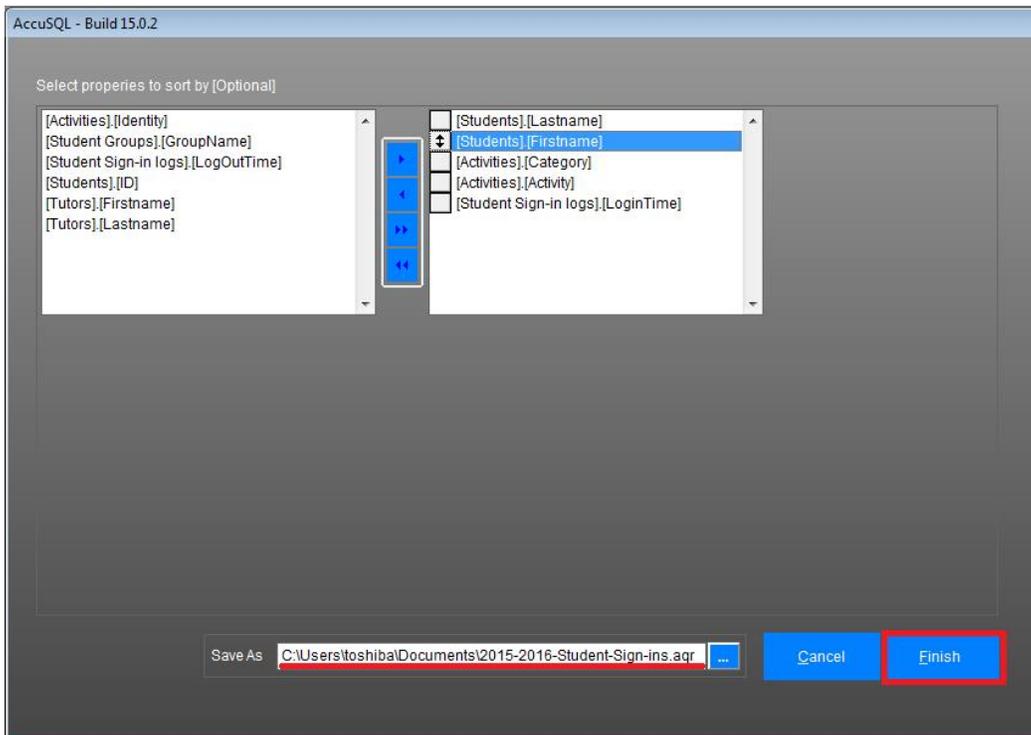
After I click Next, I will be presented with a screen where I can select how I would like to sort (and sub sort) my data. I do that by selecting a field on the left then moving it over to the right (either with the arrow or by double-clicking). After I have the fields I want on the right side, I can click on them and drag them up or down to determine my sort order. In this example, I am sorting on Student last name, then the login in time, then category, then activity.



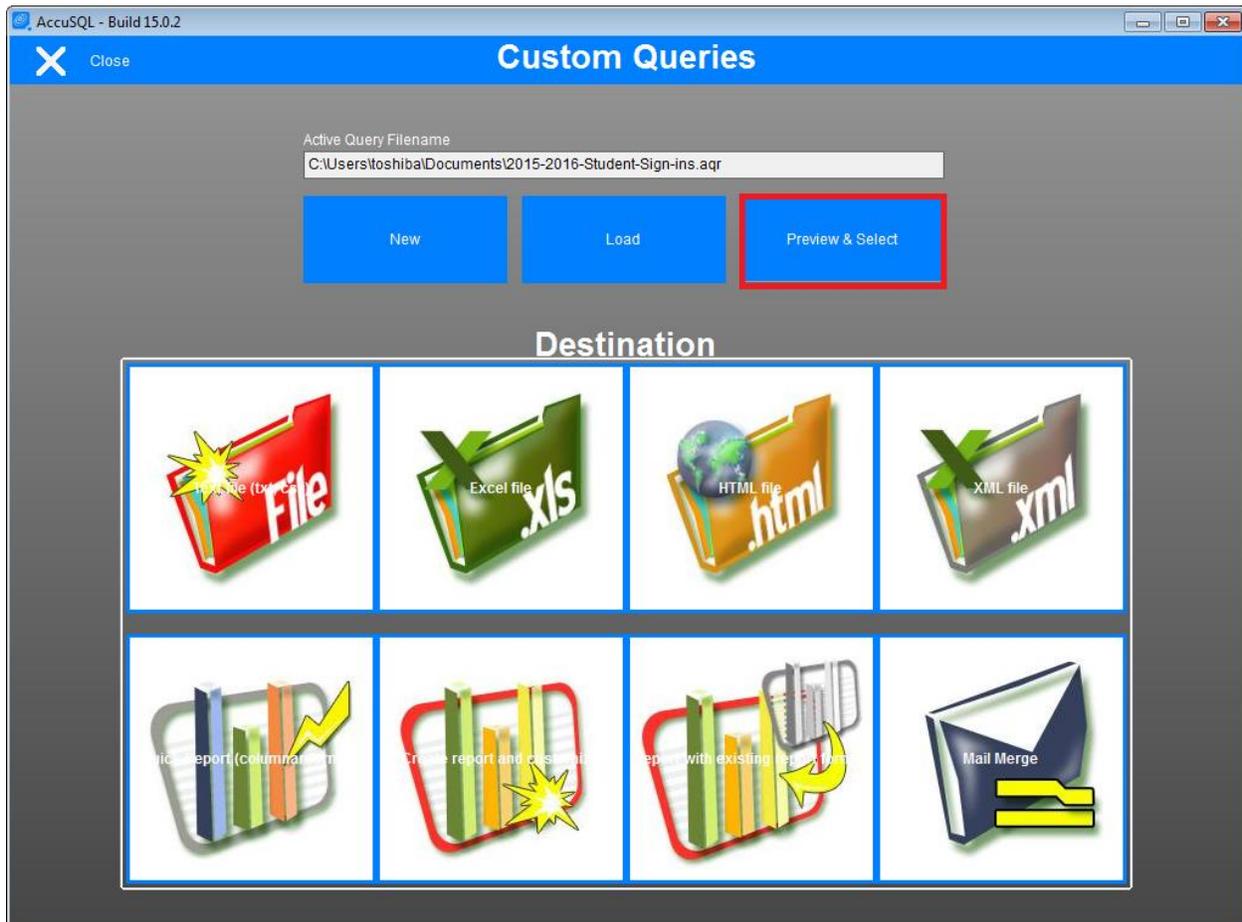
In this same screen, I can click the ... button to save my query. After I click the ... button, I am presented with a Save As screen where I can navigate to the folder where I want to save the file and also I can enter the name.



I need to click the **Open** button to actually save this query file name.



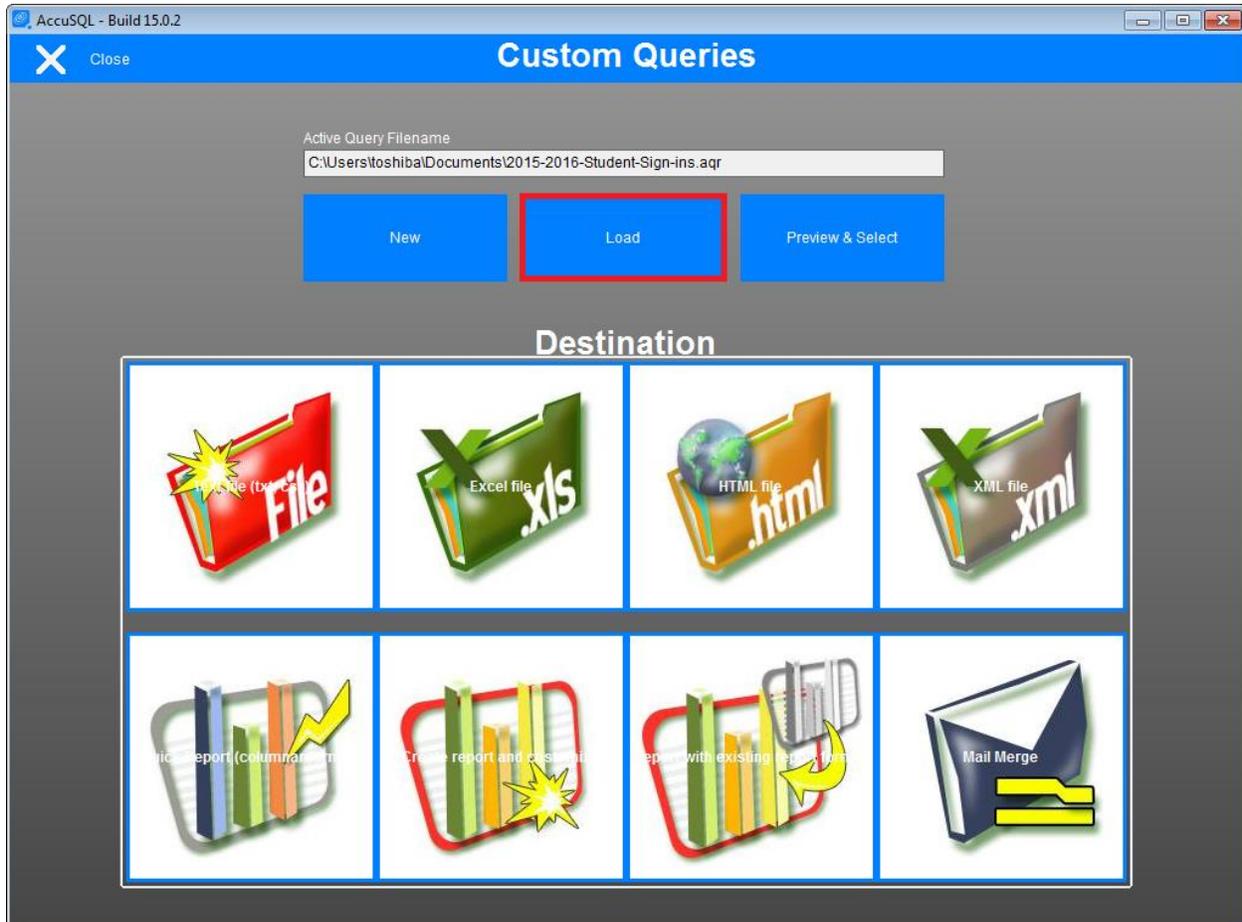
Finally, I can click the **Finish** button. I have the option here of previewing my results by pressing the **Preview & Select** button. If I do that, however, none of my records are selected, so I need to right-click in the preview window and choose Select All to highlight all my records.



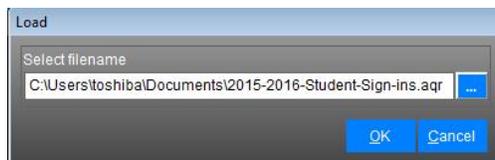
Note: If you do not press Preview & Select, all of the records are selected by default.

When I press **Close**, I am returned back to my Custom Queries window where I can choose the output format for my report. You can experiment with these to see what each report output looks like. In this case, I want to export to Excel, so I will click the Excel File .xls button. After I click the Excel .xls button, my report will automatically open in Excel.

Done! That was quite a bit of work, so I would not want to go through it all again next time I want to run that query. If I saved the query, I won't have to. All I need to do in the future is open Query Generator and click the Load button to load any queries I have already created and saved.



I click the ... button in the Load screen and then find the file on my computer where I saved it, and then click the Open button to load it back to my Load screen.



Finally, I click to OK button to load the query. I can then select any report output I desire.

Note: The query results are dynamic. That means they will change depending on the data. In this example, if I added another student to my At Risk student group and that student had signed in between 06/01/2015 and 06/01/2016, then he/she would show up in the report.