Ad Hoc Advanced

Adding Functions to Filters
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.

1. On the Field Selection page, add a function to a filter by selecting the Add Function button. The Function Editor appears in a new window.
2. Enter the following:
   a. Name
   c. The “Constant Value” field is only used with the Constant Function.
   d. Use the Filter by Search field to search for desired fields. Select which fields to include within the function by clicking on each field within the All Fields window.
3. Click “Save”
## Function Descriptions

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Description</th>
<th>Sample Function</th>
<th>Sample Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>Output the Constant Value entered on each record returned when the filter is exported.</td>
<td></td>
<td><img src="image" alt="Sample Result" /></td>
</tr>
<tr>
<td>Coalesce</td>
<td>Allows users to define multiple fields where logic pulls the first field and if NULL, the second field is pulled and so on until a value is found. Logic pulls field values in the order fields are selected in the Function Editor.</td>
<td></td>
<td><img src="image" alt="Sample Result" /></td>
</tr>
<tr>
<td>Concatenate</td>
<td>Allows field values to be appended when the filter is exported.</td>
<td></td>
<td><img src="image" alt="Sample Result" /></td>
</tr>
<tr>
<td>Add</td>
<td>Allows field values to be added together to output a single result.</td>
<td></td>
<td><img src="image" alt="Sample Result" /></td>
</tr>
<tr>
<td>Subtract</td>
<td>Allows field values to be subtracted from each other to output a single result.</td>
<td></td>
<td><img src="image" alt="Sample Result" /></td>
</tr>
<tr>
<td>Multiply</td>
<td>Allows field values to be multiplied together to output a single result.</td>
<td></td>
<td><img src="image" alt="Sample Result" /></td>
</tr>
<tr>
<td>Divide</td>
<td>Allows two or more fields to be divided and output a single result.</td>
<td></td>
<td><img src="image" alt="Sample Result" /></td>
</tr>
<tr>
<td>Record Count</td>
<td>Allows users to report a record count for the field selected.</td>
<td></td>
<td><img src="image" alt="Sample Result" /></td>
</tr>
<tr>
<td>Distinct Count</td>
<td>Allows users to report a distinct count for the field selected.</td>
<td></td>
<td><img src="image" alt="Sample Result" /></td>
</tr>
</tbody>
</table>

*Infinite Campus’s Campus Community articles for Ad Hoc Reporting were used in the development of this handout.*
## Function Descriptions

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Description</th>
<th>Sample Function</th>
<th>Sample Result</th>
</tr>
</thead>
</table>
| **MIN**       | Allows users to report the minimum value for a field. | Parameters: kyAttDailySummary.date | |}
| **MAX**       | Allows users to report the maximum value for a field. | Parameters: rosters.studentCount | |}
| **Sum**       | Adds the value or field selected over all other aggregated fields. | Parameters: transcriptDetail.creditsEarned | |}
| **Avg**       | Allows users to report the average value for a field. | Parameters: transcriptDetail.percent | |}

*Infinite Campus’s Campus Community articles for Ad Hoc Reporting were used in the development of this handout.*
Logical Expressions

The Logical Expression field allows users to incorporate conditions between fields within a filter. This field provides an effective way to use 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 can be grouped using ( ) symbols and the ID number to define the order in which the tool should include or exclude a record. In the screen shot below, the ( ) symbols indicate the tool should use the student’s primary address (2) and the student’s school number (3), include students whose city does not equal Lexington or the location code = 99999, and the county = Fayette or the county is null.

Filter the data

<table>
<thead>
<tr>
<th>ID</th>
<th>Field</th>
<th>Operator</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>mailingAddress.city</td>
<td>&lt;&gt;</td>
<td>Lexington</td>
</tr>
<tr>
<td>2</td>
<td>mailingAddress.secondary</td>
<td>= FALSE</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>sch.number</td>
<td>NOT IN</td>
<td>904,905,887,888,068</td>
</tr>
<tr>
<td>4</td>
<td>mailingAddress.county</td>
<td>&lt;&gt;</td>
<td>Fayette</td>
</tr>
<tr>
<td>5</td>
<td>mailingAddress.county</td>
<td>IS NULL</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>mailingAddress.location_code</td>
<td>=</td>
<td>99999</td>
</tr>
</tbody>
</table>

Infinite Campus’s Campus Community articles for Ad Hoc Reporting were used in the development of this handout.
Ad Hoc Advanced

Pass-Through SQL Query
Users can return custom sets of data by using the pass-through SQL query option, available as part of the Ad Hoc Filter Designer tool. A pass-through query uses SQL language to gather the desired information. A basic understanding of SQL and a working knowledge of the Campus data schema is helpful. Pass-through queries allow users to search for data in a more customized way, to search on tables and views not used in the Query Wizard and to use SQL operators.

1. Go to Index > Ad Hoc Reporting > Filter Designer
2. Select the following:
   a. Filter Type – Pass-through SQL Query
   b. Data Type – Choose as desired
3. Click “Create”
4. Select the following:
   a. Filter Name – Enter a name for the filter
   b. Short Description
   c. Long Description
   d. In the text fields on the left hand side of the screen, enter more querying definitions. The beginning part of the query is already written. Users can join two or more tables and fields here.
   e. In the second text field, enter more criteria for the SQL WHERE clause.
   f. Click the “Test Query” button to verify that the data returned is the data needed. Results will appear in the Test Query Results field on the right hand side of the screen.
   g. Select which group to “Save To”
5. Click “Save”

Sample Pass-Through SQL Queries
The following pages provide examples of commonly used pass-through queries, organized by the Filter Data Type selected for the query on the Filter Designer main page.

- “Box A” refers to the upper text box that continues the SQL statement
- “Box B” refers to the lower text box that specifies conditions of returned results

Infinite Campus’s Campus Community articles for Ad Hoc Reporting were used in the development of this handout.
**Sample Pass-Through SQL Queries - Student**

**Student Birth Date from Range**
*This query returns students whose birth dates exist within a range of time between two dates. It will include student birth dates occurring on the specified start or end date.*

**Box A**

*No text necessary*

**Box B**

\[
\text{AND student.birthdate BETWEEN 'MM/DD/YYYY' AND 'MM/DD/YYYY'}
\]

**Students without Guardians**
*This query returns students who do not have a guardian relationship assigned.*

**Box A**

\[
\text{LEFT OUTER JOIN RelatedPair rp ON rp.personID1 = student.personID AND rp.guardian = 1}
\]

**Box B**

\[
\text{AND rp.personID2 IS NULL}
\]

**Students with Multiple Guardians**
*This query returns students who have two or more guardian relationships assigned.*

**Box A**

\[
\text{INNER JOIN RelatedPair rp ON rp.personID1 = student.personID AND rp.guardian = 1}
\]

**Box B**

\[
\text{GROUP BY student.personID, student.lastName, student.firstName, student.grade, student.studentNumber}
\text{HAVING COUNT(rp.personID1) > 2}
\]

**Students who did not make a course request for a certain type of course by Course Group**
*This query returns a list of students who did not make a course request for a type of course, like Language Arts or Math. Modify the name of the course group (in the query as Language Arts) to meet the needs of your district. Ensure the query is returning the credit name, not the department name.*

**Box A**

\[
\text{left join (Request r}
\text{ join Course crs ON crs.courseID = r.courseID}
\text{ join GradingTaskCredit gtc ON gtc.courseID = crs.courseID}
\text{ join CurriculumStandard cs ON cs.standardID = gtc.standardID And cs.name = 'Language Arts') ON r.personID = student.personID And student.calendarID = crs.calendarID}
\]

**Box B**

\[
\text{And crs.number IS NULL}
\]

*Infinite Campus’s Campus Community articles for Ad Hoc Reporting were used in the development of this handout.*
Sample Pass-Through SQL Queries - Student

Students who did not make a course request for a certain type of course by Department

This query returns a list of students who did not make a course request for a type of course, like Language Arts or Math. Modify the name of the Department (in the query as Language Arts) to meet the needs of your district.

Box A

```sql
left join (Request r
join Course crs ON crs.courseID = r.courseID
join Department d ON d.departmentID = crs.departmentID And d.name = 'Language Arts') ON r.personID = student.personID
And student.calendarID = crs.calendarID
```

Box B

```sql
And crs.number IS NULL
```

Students who did not get a Requested Course

This query returns students who did not get scheduled into a course they requested. The course number must be specified.

Box A

```sql
INNER JOIN Request r ON r.personID = student.personID
And r.calendarID = student.calendarID
INNER JOIN Course c ON c.courseID = r.courseID AND c.number = '0355'
INNER JOIN Section s ON s.courseID = c.courseID
INNER JOIN Trial tr ON tr.trialID = s.trialID
And tr.active = 1
LEFT OUTER JOIN Roster ro ON ro.personID = r.personID
And ro.sectionID = s.sectionID
```

Box B

```sql
AND ro.personID IS NULL
```

Students with Fewer Credits than Desired for Grade

This query returns students enrolled in a certain grade who have less than a specified amount of credits. This example will return all twelfth-graders who have fewer than 20 credits.

Box A

```sql
INNER JOIN v_TranscriptDetail td ON td.personID = student.personID
```

Box B

```sql
AND student.grade = '12'
GROUP BY student.personID, student.lastname, student.firstname, student.grade, student.studentnumber
HAVING(SUM(td.creditsearned))< 20
```

Infinite Campus’s Campus Community articles for Ad Hoc Reporting were used in the development of this handout.
Ad Hoc Advanced

Sample Pass-Through SQL Queries - Student

Students on the A Honor Roll

This query returns students who are considered "A" honor-roll students. This query may require user modifications to fit specific needs.

Box A

BEGIN
INNER JOIN TermSchedule ts on ts.structureID = student.structureID
INNER JOIN Term t on t.termScheduleID = ts.termScheduleID and t.seq = 1
INNER JOIN GradingTask k on k.name = 'Quarter Grade'
LEFT OUTER JOIN Term tx on tx.termScheduleID = t.termScheduleID and tx.seq = 1
INNER JOIN GradingScore g on g.calendarID = student.calendarID and g.termID =
and (g.score like 'A%')
AND g.personID = student.personID
LEFT OUTER JOIN GradingScore gx on gx.calendarID = g.calendarID and gx.termID =
and gx.personID = g.personID
AND NOT (gx.score like 'A%')
END

Students on the A/B Honor Roll

This query returns students who are considered "A" or "B" honor-roll students. This query may require user modifications to fit specific needs.

Box A

BEGIN
INNER JOIN TermSchedule ts on ts.structureID = student.structureID
INNER JOIN Term t on t.termScheduleID = ts.termScheduleID and t.seq = 1
INNER JOIN GradingTask k on k.name = 'Nine Week' LEFT OUTER JOIN Term tx on

LEFT OUTER JOIN Term tx on tx.termScheduleID = t.termScheduleID and tx.seq = 1
INNER JOIN GradingScore g on g.calendarID = student.calendarID and g.termID =
and (g.score like 'A%' OR g.score like 'B%')
AND g.personID = student.personID
LEFT OUTER JOIN GradingScore gx on gx.calendarID = g.calendarID and gx.termID =
gx.personID = g.personID
AND NOT (gx.score like 'A%' OR gx.score like 'B%')
END

Box B

AND gx.scoreID IS NULL

Infinite Campus’s Campus Community articles for Ad Hoc Reporting were used in the development of this handout.
Sample Pass-Through SQL Queries - Student

Students on the A/B Honor Roll Excluding Students with only A's

This query returns students who are considered "A" or "B" honor-roll students, but excludes those who received only A's. This query may require user modifications to fit specific needs.

Box A

```
INNER JOIN TermSchedule ts on ts.structureID = student.structureID INNER JOIN Term t on t.termScheduleID = ts.termScheduleID and t.seq = 1
INNER JOIN GradingTask k on k.name = 'Quarter'
LEFT OUTER JOIN Term tx on tx.termScheduleID = t.termScheduleID and tx.seq = 1 INNER
JOIN GradingScore g on g.calendarID = student.calendarID and g.termID = t.termID and g.taskID = k.taskID and (g.score like 'A%' OR g.score like 'B%')
AND g.personID = student.personID LEFT OUTER JOIN GradingScore gx on gx.calendarID = g.calendarID and gx.termID = g.termID and gx.taskID = g.taskID and gx.personID = g.personID AND NOT (gx.score like 'A%' OR gx.score like 'B%')
INNER JOIN GradingScore g3 ON g3.calendarID = g.calendarID AND g3.termID = g.termID AND g3.taskID = g.taskID AND g3.personID = g.personID AND(g3.score LIKE 'B%')
```

Box B

```
AND gx.personID IS NULL
```

Infinite Campus’s Campus Community articles for Ad Hoc Reporting were used in the development of this handout.
Ad Hoc Advanced

Sample Pass-Through SQL Queries – Course/Section

Courses without Assigned Grading Tasks
This query returns courses to which a grading task has not yet been assigned.

Box A

```
LEFT OUTER JOIN gradingtaskcredit gtc ON gtc.courseID = course.courseID
AND gtc.calendarID = course.calendarID
```

Box B

```
AND gtc.taskCreditID IS NULL
```

Courses without Sections
This query returns courses to which a section has not yet been assigned.

Box A

```
LEFT OUTER JOIN [Section] s ON s.courseID = course.courseID
AND course.active = 1
LEFT OUTER JOIN Trial tr on tr.trialID = s.trialID AND tr.active = 1
```

Box B

```
AND s.sectionID IS NULL
```

Sample Pass-Through SQL Queries - Staff

Teachers with no Lesson Plans for the Current School Year
This query returns information on teachers who have not set up the Lesson Planner in the current school year.

Box A

```
INNER JOIN Section se ON se.teacherPersonID = individual.personID
INNER JOIN Course c ON c.courseID = se.courseID
INNER JOIN Calendar cal ON cal.calendarID = c.calendarID
INNER JOIN SchoolYear sy on sy.endyear = cal.endyear and sy.active = 1
LEFT OUTER JOIN LessonPlanGroup lpg ON lpg.sectionID = se.sectionID
LEFT OUTER JOIN LessonPlanGroupActivity lpga ON lpga.groupID = lpg.groupID
LEFT OUTER JOIN LessonPlanActivity act ON act.activityID = lpga.activityID
INNER JOIN activeTrial at ON at.trialID = se.trialID
```

Box B

```
GROUP BY individual.personID, individual.lastName, individual.firstName HAVING
SUM(COALESCE(act.acti
```
Commonly used codes and descriptions

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Hispanic/Latino</td>
</tr>
<tr>
<td>2</td>
<td>American Indian or Alaska Native</td>
</tr>
<tr>
<td>3</td>
<td>Asian</td>
</tr>
<tr>
<td>4</td>
<td>Black or African American</td>
</tr>
<tr>
<td>5</td>
<td>Native Hawaiian or other Pacific Islander</td>
</tr>
<tr>
<td>6</td>
<td>White</td>
</tr>
<tr>
<td>7</td>
<td>Two or more races</td>
</tr>
</tbody>
</table>

Infinite Campus’s Campus Community articles for Ad Hoc Reporting were used in the development of this handout.