Data Days & Master Scheduling: Data Pulling Contest

Think you build amazing Ad Hoc Queries? How about writing excellent Pass-through SQL Queries or SQL select statements? Well this is your chance to prove it, help others along the way, and even win a prize for your efforts!

We are holding a contest for the "most useful" Ad Hoc Query Wizard Report, Pass-through SQL Query, and SQL Select Statement. (We are looking for the queries that you find helpful in running on a regular basis, or queries that help solve a common and or complex request for data) All entries will be voted on by attendees of the Data Days & Master Scheduling Event, with winners announced on Thursday, March 4. Individuals can submit up to 3 entries per category. If there are duplicate queries or queries similar enough, the query selected for the contest will be recorded as the first person to enter their submission.

Entries should be sent to Training@InfiniteCampus.com

Categories

1. Ad Hoc Query Wizard
2. Pass-Through SQL Query
3. SQL Select Statement

What to submit

Ad Hoc Query Wizard:

Include a description of the purpose of the query and the type of data that it is meant to pull. In addition, include screen shots of the following

- The Selected Fields
- Any operators / logical expressions used to narrow the scope of your data (if applicable)
- Any groupings / aggregations you find helpful (if applicable)

Pass-Through SQL Query:

Include a description of what the Query is designed to look for, along with text for Box A and Box B. Highlight the parameters that can be modified to adjust the scope of the data being pulled.

SQL Select Statement:

Include a description of what the statement is designed to pull, along with the text of your Select Statement that can be used in the Infinite Campus Database. Highlight the parameters that can be modified to adjust the scope of the data being pulled

Examples for each category can be found on the following pages.

Ad Hoc Query Wizard Example Submission:

This Query Wizard Filter pulls students in specified grade levels and it allows me to use it when running other reports and processes within Infinite Campus.

Selected Fields:

Selected Fields
student.studentNumber
student.lastName
student.firstName
student.grade
student.activeToday

Filter Parameters:

Filter the data

	ID *Field	Operator	Value
✕	1 student.studentNumber		
✕	2 student.lastName		
✕	3 student.firstName		
✕	4 student.grade	IN	01, 02, 03
✕	5 student.activeToday	= TRUE	

Add

Logical Expression (Optional):

Grouping and Aggregation

Group the data into sections that can have aggregates/sub-totals

Grouping	Group by	Group Order
Tier 1	student.grade	Ascending
Tier 2		Ascending
Tier 3		Ascending
Tier 4		Ascending
Tier 5		Ascending

Aggregate/Sub Total by	Aggregate Type
student.studentNumber	Record Count

4321 109th Avenue NE || Blaine, MN 55449 || 800 850 2335 || infinitecampus.com

Pass-Through SQL Query Example Submission:

Description

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:

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

Box B:

```
AND student.grade = '12' GROUP BY student.personID, student.lastname, student.firstname, student.grade,
student.studentnumber

HAVING(SUM(td.creditsearned))< 20
```

SQL Select Statement Example Submission:

This select statement is meant to find students that have requested one course, but not another when they need to have both courses requested. In this query, the results will find when a student has a request for the second course, but not the first. You can change the highlighted r1 and r2 to get the opposite result

```
Select student.lastname, student.firstname, c1.name, c2.name
From Student
INNER JOIN Course c1 ON c1.calendarID = student.calendarID AND c1.number = '4100'

LEFT OUTER JOIN Request r1 ON r1.personID = student.personID
AND r1.courseID=c1.courseID

INNER JOIN Course c2 ON c2.calendarID = student.calendarID AND c2.number = '4150'
LEFT OUTER JOIN Request r2 ON
r2.personID = student.personID AND r2.courseID=c2.courseID

Where 1 = 1
AND student.calendarid = 250
AND ((r1.requestID IS NULL
AND r2.requestID IS NOT NULL))
```

Please submit contest entries to Training@InfiniteCampus.com