

# Student Pass-Through Queries

June 2015

**This document is intended for restricted use only.**

**Infinite Campus asserts that this document contains proprietary information that would give our competitors undue advantage should they come into possession of any part or all of it. As such, this document cannot be publicly disclosed unless so ordered by a court of competent jurisdiction.**

©2015 Infinite Campus, Inc. All rights reserved.

INFINITE CAMPUS and Transforming K12 Education are registered trademarks of Infinite Campus, Inc. The INFINITE CAMPUS logo is a trademark of Infinite Campus, Inc. This publication, or any part thereof, may not be reproduced or transmitted in any form or any means, electronic or mechanical, including photocopying, recording, storage in an information retrieval system, or otherwise, by anyone other than Infinite Campus, Inc. without written permission of Infinite Campus, Inc., 4321 109th Avenue NE, Blaine, MN 55449, tel. (651) 631-0000, email [info@infinitecampus.com](mailto:info@infinitecampus.com).

# Student Pass-Through Queries

**Simulation****Video**

[Student Queries for Census and Demographic Data](#) | [Student Queries for Course Information](#) | [Student Queries for Academic Information](#) | [Student Queries for Behavior Information](#) | [Miscellaneous Student Queries](#)

**PATH:** *Ad hoc Reporting > Filter Designer > Pass-Through SQL Query*

Student pass-through queries inherit the year, school, calendar and schedule selected in the Campus toolbar.

## Student Queries for Census and Demographic Data

### 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

```
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

```
LEFT OUTER JOIN RelatedPair rp ON rp.personID1 = student.personID AND  
rp.guardian = 1
```

#### Box B

```
AND rp.personID2 IS NULL
```

## Students with Multiple Guardians

*This query returns students who have two or more guardian relationships assigned.*

### Box A

```
INNER JOIN RelatedPair rp ON rp.personID1 = student.personID
AND rp.guardian = 1
```

### Box B

```
GROUP BY student.personID, student.lastName, student.firstName, student.grade,
student.studentNumber
HAVING COUNT(rp.personID1) > 2
```

## Students without a Current Household

*This query returns students who do not have a current household indicated.*

### Box A

```
LEFT OUTER JOIN HouseholdMember hm ON hm.personID = student.personID
AND (hm.endDate IS NULL OR hm.endDate > getDate())
```

### Box B

```
AND hm.personID IS NULL
```

## Students in a Household by Themselves

*This query returns students who are in a household with no other members.*

### Box A

```
INNER JOIN HouseholdMember hm1 ON hm1.personID = student.personID
AND (hm1.startDate IS NULL OR hm1.startDate <= getDate()) AND (hm1.endDate IS
NULL OR hm1.endDate > getDate())
INNER JOIN HouseholdMember hm2 ON hm2.householdID = hm1.householdID
AND (hm2.startDate IS NULL OR hm2.startDate <= getDate())AND (hm2.endDate IS
NULL OR hm2.endDate > getDate())
```

### Box B

```
GROUP BY student.personID, student.lastname, student.firstname, student.grade,  
student.studentnumber  
HAVING COUNT(hm2.memberID) = 1
```

## Students in Households with a Combined Total of More than Five

*This query returns students who are in a household of more than five members.*

### Box A

```
INNER JOIN HouseholdMember hm1 ON hm1.personID = student.personID  
AND (hm1.startDate IS NULL OR hm1.startDate <= getDate()) AND (hm1.endDate IS  
NULL OR hm1.endDate > getDate ())  
INNER JOIN HouseholdMember hm2 ON hm2.householdID = hm1.householdID  
AND (hm2.startDate IS NULL OR hm2.startDate <= getDate ()) AND (hm2.endDate IS  
NULL OR hm2.endDate > getDate())
```

### Box B

```
GROUP BY student.personID, student.lastname, student.firstname, student.grade,  
student.studentnumber, hm1.householdID  
HAVING COUNT(hm2.memberID) > 5
```

## Students in a Household with no Address

*This query returns students who have no address assigned to their household(s).*

### Box A

```
INNER JOIN HouseholdMember HM ON HM.personID=student.personID  
INNER JOIN household H on h.householdID=HM.householdID  
LEFT OUTER JOIN HouseholdLocation L on L.householdID=H.householdID
```

### Box B

```
AND L.addressID IS NULL
```

## Student not in any Household with a Mailing Address - Includes Secondary Addresses

*This query returns students who are not in any household with a mailing address. If a student has three addresses, where one of them is marked as a mailing address but the other two are not, the student will **NOT** be returned by the query.*

### Box A

```
LEFT OUTER JOIN HouseholdMember hm ON hm.personID = student.personID AND
(hm.endDate IS NULL OR hm.endDate > getDate())
LEFT OUTER JOIN HouseholdLocation hl ON hl.householdID = hm.householdID
AND (hl.endDate IS NULL OR hl.endDate > getDate())
AND hl.mailing = 1
```

### Box B

```
AND hl.locationID IS NULL
```

## Students not in any Household with a Mailing Address - Excludes Secondary Household and Secondary Addresses

*This query returns students who are not in any household with a mailing address. If a student has three addresses, where one of them is marked as a mailing address but the other two are not, the student will **NOT** be returned by the query.*

### Box A

```
LEFT OUTER JOIN HouseholdMember hm ON hm.personID = student.personID AND
(hm.endDate IS NULL OR hm.endDate > getDate())
AND (hm.secondary IS NULL OR hm.secondary = 0)
LEFT OUTER JOIN HouseholdLocation hl ON hl.householdID = hm.householdID
AND (hl.endDate IS NULL OR hl.endDate > getDate())
AND (hl.secondary IS NULL OR hl.secondary = 0) AND hl.mailing = 1
```

### Box B

```
AND hl.locationID IS NULL
```

## Students with More Than One Primary Mailing Address

*This query returns students who have more than one primary mailing address listed.*

**Box A**

```
INNER JOIN HouseholdMember hm ON hm.personID = student.personID
AND (hm.endDate IS NULL OR hm.endDate > getDate()) AND (hm.secondary IS NULL
OR hm.secondary = 0)
INNER JOIN HouseholdLocation hl ON hl.householdID = hm.householdID
AND (hl.endDate IS NULL OR hl.endDate > getDate())AND (hl.secondary IS NULL OR
hl.secondary = 0) AND hl.mailing = 1
```

**Box B**

```
GROUP BY student.personID, student.lastName, student.firstName, student.grade,
student.studentNumber HAVING COUNT(hl.locationID) > 1
```

**Students who do not have at least one parent/guardian with a portal account**

*This query returns students who do not have a parent/guardian who has a Campus Portal account.*

**Box A**

```
LEFT JOIN (RelatedPair rp
JOIN HouseholdMember hmG ON hmG.personID = rp.personID2 And hmG.endDate IS
NULL
JOIN HouseholdMember hmS ON hmS.personID = rp.personID1 And hmS.householdID =
hmG.householdID
JOIN UserAccount ua ON ua.personID = rp.personID2) ON rp.personID1 =
student.personID And rp.portal = 1 And rp.guardian = 1
```

**Box B**

```
And (rp.personID1 IS NULL)
```

**Student Queries for Course Information****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**

```

left join (Request r
join Course crs ON crs.courseID = r.courseID
join GradingTaskCredit gtc ON gtc.courseID = crs.courseID
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**

```

And crs.number IS NULL

```

## 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**

```

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**

```

And
crs.number IS NULL

```

## Students Taking a Course

*This query returns students who are taking a course, as specified. Use Box B to indicate a specific course or a general type of course (e.g., specifying "%Literature%" pulls all course names containing the word "literature").*

**Box A**

```

INNER JOIN Roster r ON r.personID = student.personID
INNER JOIN Section s ON s.sectionID = r.sectionID
INNER JOIN Course c ON c.courseID = s.courseID
AND c.calendarID = student.calendarID
INNER JOIN Trial t ON t.trialID = s.trialID AND t.active = 1

```

**Box B**

```

AND c.name LIKE '%Literature%'

```

**Students Registered for Part of a Course**

*This query returns students who have requested only one semester of a year-long course, instead of both semesters (e.g., if Algebra is a year-long course, students registered for Semester 1 but not Semester 2, or vice versa, will be returned). The course number must be specified.*

**Box A**

```

INNER JOIN Course c1 ON c1.calendarID = student.calendarID AND c1.number =
'1300'
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 =
'1350'
LEFT OUTER JOIN Request r2 ON r2.personID = student.personID
AND r2.courseID=c2.courseID

```

**Box B**

```

AND ((r1.requestID IS NULL
AND r2.requestID IS NOT NULL)
OR (r2.requestID IS NULL AND r1.requestID IS NOT 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**

```

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**

```

AND ro.personID IS NULL

```

**Students who have Less Than X Requested Units**

*This query will return students who have less than a certain number of requested units.*

**Box A**

```

INNER JOIN Request r ON r.personID = student.personID
INNER JOIN Course c ON c.courseID = r.courseID

```

**Box B**

```

AND r.type IN('R','E')
AND r.calendarID = student.calendarID
GROUP by student.personID, student.lastName, student.firstName, student.grade,
student.studentNumber
HAVING(SUM(c.terms*c.schedules*c.periods))< 48

```

**Students Who Have More Than One Course In a Period**

*This query returns students who are scheduled into more than one course in a period. This only works before the start of the calendar.*

**Box A**

```

INNER JOIN Roster ro ON ro.personID = student.personID
INNER JOIN Section se ON se.sectionID = ro.sectionID
INNER JOIN Course co ON co.courseID = se.courseID AND co.calendarID =
student.calendarID
INNER JOIN SectionPlacement sp ON sp.sectionID = se.sectionID
INNER JOIN Period p ON p.periodID = sp.periodID
INNER JOIN PeriodSchedule ps ON ps.periodScheduleID = p.periodScheduleID
INNER JOIN Trial tr ON tr.trialID = se.trialID AND tr.active = 1

```

**Box B**

```

GROUP BY student.personID, student.lastName, student.firstName, student.grade,
student.studentNumber, p.periodID, ps.periodScheduleID, sp.termID
HAVING COUNT(se.sectionID) > 1

```

**Students not Scheduled into a Course**

*This query will return students who are not scheduled in a specified course.*

**Box A**

```

LEFT OUTER JOIN view_roster v on v.personID = student.personID and
v.calendarID = student.calendarID
AND v.class_num LIKE 'PE%'

```

**Box B**

```

AND v.personid IS NULL

```

**Student Queries for Academic Information****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**

```

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
```

## 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

```
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 =  
t.termID and g.taskID = k.taskID  
and (g.score like 'A%')  
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%')
```

### Box B

```
AND gx.scoreID IS NULL
```

## 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

```

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
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.scoreID IS NULL

```

**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
```

## Student Queries for Behavior Information

### Students without Behavior Events

*This query returns students who do not have behavior event records in the selected calendar.*

#### Box A

```
LEFT OUTER JOIN v_BehaviorDetail be ON student.personID = be.personID and  
student.calendarID = be.calendarID
```

#### Box B

```
AND BE.eventID IS NULL
```

### Students without Behavior Events in Date Range

*This query returns students who do not have behavior event records in the selected calendar and date range.*

#### Box A

```
LEFT OUTER JOIN v_BehaviorDetail bd ON bd.personID = student.personID AND  
bd.calendarID = student.calendarID AND bd.incidentDate BETWEEN 'MM/DD/YYYY'  
AND 'MM/DD/YYYY'
```

#### Box B

```
AND bd.eventID IS NULL
```

### Students without Behavior Events and Students with Events where Role is Not Offender or Participant

*This query returns students who do not have behavior events. It also returns students who have events in which they are not marked as an "offender" or as a "participant."*

#### Box A

```
LEFT OUTER JOIN v_BehaviorDetail bd ON bd.personID = student.personID AND  
bd.calendarID = student.calendarID
```

**Box B**

```
AND bd.personID IS NULL OR (bd.role <> 'Offender'  
AND bd.role <> 'Participant')
```

**Students without Behavior Resolutions**

*This query returns students who are involved in behavior events without resolutions.*

**Box A**

```
INNER JOIN v_BehaviorDetail bd ON bd.personID = student.personID AND  
bd.calendarID = student.calendarID
```

**Box B**

```
AND bd.resolutionID IS NULL
```

**Students with no Behavior Resolutions in Date Range**

*This query pulls students who do not have any behavior resolutions within the specified date range.*

**Box A**

```
LEFT OUTER JOIN v_BehaviorDetail bd ON bd.personID = student.personID  
AND bd.calendarID = student.calendarID  
AND bd.resolutionStartDate BETWEEN 'MM/DD/YYYY'  
AND 'MM/DD/YYYY'
```

**Box B**

```
AND bd.resolutionID IS NULL
```

## Miscellaneous Student Queries

### Finding Students Not Assigned Fees

*This query returns all students who have no fees assigned.*

#### Box A

```
LEFT JOIN feeassignment fa on student.personID = fa.personID
```

#### Box B

```
AND fa.personID is null
```

### Students with a Fee Balance for Any Year

*This query returns students who owe fees for any school year.*

#### Box A

```
join v_FeeSummary fs on fs.personID = student.personID
```

#### Box B

```
AND fs.totalBalance <0
```

### List of Students who have a Free/Reduced Eligibility

*This query returns a list of students who have a free or reduced eligibility.*

#### Box A

```
INNER JOIN v_POspatron p ON p.personID = student.personID  
INNER JOIN v_POSEligibilityCurrent pe on pe.personID = p.personID  
LEFT OUTER JOIN POSEligibility pos ON pos.personID = pe.personID
```

#### Box B

```
AND pos.eligibility IN ('R', 'F')
```

### List of Students who do NOT have a Free/Reduced Eligibility

*This query returns a list of students who do not have a free or reduced eligibility.*

#### Box A

```
LEFT OUTER JOIN POSEligibility pe ON pe.personid = student.personid
```

#### Box B

```
AND pe.eligibility IS NULL
```

### A Student's Eligibility on a Certain Day

*This query returns a student's eligibility on a certain day.*

#### Box A

```
INNER JOIN v_POSpatron p ON p.personID = student.personID  
INNER JOIN v_POSEligibilityCurrent pe on pe.personID = p.personID
```

#### Box B

```
AND p.type = 'student'  
AND (pe.elig_endDate <= GETDATE() OR pe.elig_endDate IS NULL)  
ORDER BY student.lastname
```

To pull a specific date, change GETDATE to the desired date with single quotes (i.e., '05-23-2011')

### Students without a Transportation Code

*This query returns students who do not have a transportation code.*

#### Box A

```
LEFT OUTER JOIN Transportation t ON t.personID = student.personID AND  
t.calendarID = student.calendarID
```

#### Box B

```
AND t.personID IS NULL
```

## Students with Perfect Attendance

*This query returns students who have perfect attendance records – that is, students who have no attendance marks.*

### Box A

```
LEFT OUTER JOIN attendance a ON a.personID = student.personID AND a.calendarID  
= student.calendarID
```

### Box B

```
AND a.attendanceID IS NULL
```

## Impact Aid Employment

*This query returns information on students whose parents are employed at Impact Aid sites.*

### Box A

```
INNER JOIN ImpactAidEmployment i on i.personID = student.personID
```

### Box B

```
AND i.siteID is NOT NULL  
AND student.endYear = 2010
```

## Students Assigned to a Particular Counselor

*This query returns students who are assigned to a particular counselor.*

### Box A

```
INNER JOIN TeamMember tm ON tm.personID = student.personID AND tm.module =  
'counseling' AND tm.role = 'counselor'  
AND (tm.endDate IS NULL OR tm.endDate >= getDate())  
INNER JOIN individual i ON i.personID = tm.staffPersonID
```

### Box B

```
AND i.lastName = 'Abra' AND i.firstName = 'Dean'
```

## Students whose Food Service Eligibility Code does not Match their Economic Indicator

*This query returns students whose Food Service Eligibility status does not match the Economic Indicator assigned on their enrollment record.*

*In this example, Code 0 = Ineligible; 1 = Reduced Eligible; 2 = Free Eligible.*

### Box A

```
INNER JOIN Enrollment e ON e.enrollmentID = student.enrollmentID
AND (e.endDate IS NULL OR e.endDate >= getDate())
LEFT OUTER JOIN POSEligibility pos ON pos.personID = student.personID AND
pos.endDate = student.endDate
```

### Box B

```
AND ((e.mealStatus = '0' AND pos.eligibility <> 'S')
OR (e.mealStatus = '1' AND pos.eligibility <> 'R')
OR (e.mealStatus = '2' AND pos.eligibility <> 'F')
OR (e.mealStatus IS NULL AND pos.eligibility IS NOT NULL)
OR (e.mealStatus IS NOT NULL AND pos.eligibility IS NULL))
```

## Students Attending School Outside School Boundaries

*This query returns information on students who attend a school located outside of the school boundary related to their address.*

### Box A

```
INNER JOIN School sc ON sc.schoolID = student.schoolID
INNER JOIN HouseholdMember hm on hm.personID = student.personID
AND (hm.secondary IS NULL OR hm.secondary = 0)
AND hm.endDate IS NULL
INNER JOIN HouseholdLocation hl ON hl.householdID = hm.householdID
AND (hl.secondary IS NULL or hl.secondary = 0)
AND hl.endDate IS NULL
INNER JOIN Address a ON a.addressID = hl.addressID
INNER JOIN SchoolBoundary sb ON sb.addressID = a.addressID AND sb.schoolID <>
student.schoolID
```

### Box B

No text necessary