How do I include filtered rowcounts from two other tables in a query of a third?

Asked 15 years, 11 months ago Modified 15 years, 11 months ago Viewed 122 times

1

I have a MySql database with three tables I need to combine in a query: schedule, enrolled and waitlist. They are implementing a basic class enrollment system.

Schedule contains the scheduled classes. When a user enrolls in a class, their user account id and the id of the scheduled class are stored in enrolled. If a class is at capacity, they are stored in waitlist instead. All three tables share a scheduleId column which identifies each class.

When I query the schedule table, I need to also return enrolled and waitlist columns that represent the number of users enrolled and waiting for that particular scheduleId.

A preliminary query I came up with to accomplish this was:

```
select s.id, s.classDate, s.instructor, COUNT(e.id) as enrolled
from schedule as s
left outer join enrolled as e
on s.id = e.scheduleId
group by s.id
```

which works ok for one or the other, but obviously I can't get the values for both the enrolled and waitlist tables this way. Can anybody suggest a good way of doing this?

sql mysql rowcount

Share Improve this question Follow

asked Jan 22, 2009 at 2:55

Adam Lassek

35.5k ● 14 ● 93 ● 109

3 Answers

Sorted by: Highest score (default)

4

Use nested SELECT queries. Assuming a bit about your schema, how about something like this (might not work on some flavors of SQL):

```
select s.id, s.classDate, s.instructor,
       (select COUNT(e.id) from enrolled e where e.scheduleId = s.id) as
enrolled,
       (select COUNT(w.id) from waitlist w where w.scheduleId = s.id) as
waiting
from schedule as s
group by s.id
```

Share Improve this answer Follow

answered Jan 22, 2009 at 3:16

lc.

116k ● 20 ● 161 ● 188

3

I would do it with another left join and two inline count(distincts)

```
select s.id, s.classDate, s.instructor ,
count(distinct e.id) as enrolled,
count(distinct w.id) as waiting
from schedule as s
left outer join enrolled as e
on s.id = e.scheduleID
left outer join waitlist as w
on s.id = w.scheduleID
group by s.id
```

When I ran this approach versus the subqueries it executed about twice as fast, but I am looking at a pretty small result set.

Share Improve this answer Follow

answered Jan 22, 2009 at 4:56

cmsjr

59k ● 11 ● 74 ● 63

0

Two quick ways:

1- Use `COUNT(DISTINCT e.id), COUNT(DISTINCT w.id)` to get the number of unique instances in each table, then join on both. This is possibly hideously inefficient.

2- Use subqueries in the `FROM` clause (only works in MySQL 5.0 and later):

```
SELECT s.id, s.classDate, s.instructor, tmpE.c AS enrolled, tmpW.c AS waiting
FROM
  schedule AS s,
  ( SELECT scheduleID, COUNT(*) AS c FROM enrolled GROUP BY scheduleID ) AS
tmpE,
  ( SELECT scheduleID, COUNT(*) AS c FROM waiting GROUP BY scheduleID ) AS tmpW
WHERE
    s.id = e.scheduleID
    AND s.id = w.scheduleID
GROUP BY s.id
```

I may have missed a left join in there, though.

Share Improve this answer Follow

answered Jan 22, 2009 at 3:10

kyle

1,470 ● 2 ● 15 ● 21