SQL Query Help - Scoring Multiple Choice Tests

Asked 16 years, 4 months ago Modified 16 years, 2 months ago

Viewed 2k times

▲

0

▼

Say I have a Student table, it's got an int ID. I have a fixed set of 10 multiple choice questions with 5 possible answers. I have a normalized answer table that has the question id, the Student.answer (1-5) and the Student.ID

I'm trying to write a single query that will return all scores over a certain pecentage. To this end I wrote a simple UDF that accepts the Student.answers and the correct answer, so it has 20 parameters.

I'm starting to wonder if it's better to denormalize the answer table, bring it into my applcation and let my application do the scoring.

Anyone ever tackle something like this and have insight?

dynamic-sql

Share

Improve this question

Follow

edited Sep 26, 2008 at 19:46

Chris

6,842 ● 6 ● 53 ● 67

asked Aug 22, 2008 at 13:53

Webjedi

4,737 ● 7 ● 43 ● 59

6 Answers

Sorted by: Highest score (default)

▲

2

▼

✓

If I understand your schema and question correctly, how about something like this:

```
select student_name, score
from students
  join (select student_answers.student_id,
count(*) as score
        from student_answers, answer_key
        group by student_id
        where student_answers.question_id =
answer_key.question_id
          and student_answers.answer =
answer_key.answer)
  as student_scores on students.student_id =
student_scores.student_id
where score >= 7
order by score, student_name
```

That should select the students with a score of 7 or more, for example. Just adjust the where clause for your purposes.

Share Improve this answer Follow

answered Aug 22, 2008 at 14:11

Dave Ward

60.5k ● 14 ● 118 ● 134

▲

I would probably leave it up to your application to perform the scoring. Check out [Maybe Normalizing Isn't Normal](#)

1

by Jeff Atwood.

Share Improve this answer

Follow

answered Aug 22, 2008 at 13:58

Bryan Roth

10.7k ● 15 ● 49 ● 56

0

The architecture you are talking about could become very cumbersome in the long run, and if you need to change the questions it means more changes to the UDF you are using.

I would think you could probably do your analysis in code without necessarily de-normalizing your database. De-normalization could also lend to inflexibility, or at least added expense to update, down the road.

Share Improve this answer

Follow

answered Aug 22, 2008 at 14:00

palehorse

27.4k ● 4 ● 42 ● 49

0

No way, you definitely want to keep it normalized. It's not even that hard of a query.

Basically, you want to left join the students correct answers with the total answers for that question, and do a count. This will give you the percent correct. Do that for each student, and put the minimum percent correct in a where clause.

Share Improve this answer

Follow

answered Aug 22, 2008 at 14:01

Shawn

19.8k ● 20 ● 100 ● 153

0

Denormalization is generally considered a last resort. The problem seems very similar to survey applications, which are very common. Without seeing your data model, it's difficult to propose a solution, but I will say that it is definitely possible. I'm wondering why you need 20 parameters to that function?

A relational set-based solution will be simpler and faster in most cases.

Share Improve this answer

Follow

answered Aug 22, 2008 at 14:02

Eric Z Beard

38.4k ● 27 ● 101 ● 147

0

This query should be quite easy... assuming you have the correct answer stored in the question table. You do have the correct answer stored in the question table, right?

Share Improve this answer

Follow

answered Aug 22, 2008 at 14:28

Stu

15.8k ● 4 ● 45 ● 74