

SQL query to get the top "n" scores out of a list

Asked 16 years, 3 months ago Modified 7 years, 5 months ago Viewed 25k times



7

I'd like to find the different ways to solve a real life problem I had: imagine to have a contest, or a game, during which the users collect points. You have to build a query to show the list of users with the best "n" scores.



I'm making an example to clarify. Let's say that this is the Users table, with the points earned:



```
UserId - Points
1      - 100
2      - 75
3      - 50
4      - 50
5      - 50
6      - 25
```

If I want the top 3 scores, the result will be:

```
UserId - Points
1      - 100
2      - 75
3      - 50
4      - 50
5      - 50
```

This can be realized in a view or a stored procedure, as you want. My target db is Sql Server. Actually I solved this, but I think there are different way to obtain the result... faster or more efficient than mine.

sql

sql-server

puzzle

Share

Improve this question

Follow

edited Jun 5, 2015 at 6:01



sqluser

5,652 ● 7 ● 42 ● 52

asked Sep 1, 2008 at 11:03



ila

4,724 ● 7 ● 38 ● 41

11 Answers

Sorted by: Highest score (default)



Untested, but should work:

11

```
select * from users where points in
(select distinct top 3 points from users order by points desc)
```

Share Improve this answer Follow

answered Sep 1, 2008 at 11:07



Espo

41.9k ● 21 ● 136 ● 161

Here's one that works - I don't know if it's more efficient, and it's SQL Server 2005+

4

```
with scores as (
    select 1 userid, 100 points
    union select 2, 75
    union select 3, 50
    union select 4, 50
    union select 5, 50
    union select 6, 25
),
results as (
    select userid, points, RANK() over (order by points desc) as ranking
    from scores
)
select userid, points, ranking
from results
where ranking <= 3
```

Obviously the first "with" is to set up the values, so you can test the second with, and final select work - you could start at "with results as..." if you were querying against an existing table.

Share Improve this answer Follow

answered Sep 1, 2008 at 11:59



crucible

3,137 ● 2 ● 28 ● 35

I have a similar issue and was trying to use MAX and then I read your answer and remembered DENSE_RANK. Saved me a lot of time. – [DataGirl](#) Jun 21, 2012 at 20:49

How about:

1

```
select top 3 with ties points
from scores
order by points desc
```

Not sure if "with ties" works on anything other the SQL Server.



On SQL Server 2005 and up, you can pass the "top" number as an int parameter:

```
select top (@n) with ties points
from scores
order by points desc
```

Share

edited Sep 1, 2008 at 12:09

answered Sep 1, 2008 at 11:32

Improve this answer

Follow



[Matt Hamilton](#)

204k ● 61 ● 392 ● 321



1



Actually a modification to the WHERE IN, utilizing an INNER JOIN will be much faster.

```
SELECT
    userid, points
FROM users u
INNER JOIN
(
    SELECT DISTINCT TOP N
        points
    FROM users
    ORDER BY points DESC
) AS p ON p.points = u.points
```

Share Improve this answer Follow

answered Sep 18, 2008 at 1:57



[Chris Lampley](#)



0



@bosnic, I don't think that will work as requested, I'm not that familiar with MS SQL but I would expect it to return only 3 rows, and ignore the fact that 3 users are tied for 3rd place.

Something like this should work:

```
select userid, points
from scores
where points in (select top 3 points
                  from scores
                  order by points desc)
order by points desc
```

Share Improve this answer Follow

answered Sep 1, 2008 at 11:12



[Tom](#)

44.4k ● 35 ● 99 ● 102

@Rob#37760:



```
select top N points from users order by points desc
```

0

This query will only select 3 rows if N is 3, see the question. "Top 3" should return 5 rows.



Share Improve this answer Follow

answered Sep 1, 2008 at 11:21



Espo

41.9k ● 21 ● 136 ● 161



@Espo thanks for the reality check - added the sub-select to correct for that.

0

I think the easiest response is to:



```
select userid, points from users
where points in (select distinct top N points from users order by points desc)
```



If you want to put that in a stored proc which takes N as a parameter, then you'll either have to do read the SQL into a variable then execute it, or do the row count trick:



```
declare @SQL nvarchar(2000)
set @SQL = "select userID, points from users "
set @SQL = @SQL + " where points in (select distinct top " + @N
set @SQL = @SQL + " points from users order by points desc)"

execute @SQL
```

or

```
SELECT UserID, Points
FROM (SELECT ROW_NUMBER() OVER (ORDER BY points DESC)
AS Row, UserID, Points FROM Users)
AS usersWithPoints
WHERE Row between 0 and @N
```

Both examples assume SQL Server and haven't been tested.

Share

edited Sep 1, 2008 at 11:31

answered Sep 1, 2008 at 11:15

Improve this answer



Rob Allen

17.7k ● 6 ● 53 ● 70

Follow



@Matt Hamilton

0

Your answer works with the example above but would not work if the data set was 100, 75, 75, 50, 50 (where it would return only 3 rows). TOP WITH TIES only includes



the ties of the last row returned...



Share Improve this answer Follow



answered Sep 1, 2008 at 13:02



Marius

2,542 ● 6 ● 31 ● 43



Crucible got it (assuming SQL 2005 is an option).

0

Share Improve this answer Follow

answered Sep 1, 2008 at 13:08



NakedBrunch

49.4k ● 14 ● 74 ● 98



Hey I found all the other answers bit long and inefficient My answer would be:

0

```
select * from users order by points desc limit 0,5
```

this will render top 5 points



Share Improve this answer Follow

answered Nov 1, 2013 at 6:34



Krishna Gupta

1,299 ● 11 ● 12



Try this

0

```
select top N points from users order by points desc
```



Share

Improve this answer

Follow

edited Nov 25, 2015 at 10:46



Sabyasachi Mishra

1,749 ● 2 ● 32 ● 51

answered Sep 19, 2008 at 10:03



kedar kamthe

8,178 ● 11 ● 37 ● 46