

# Best approach for sortable table with a lot of data

Asked 16 years, 3 months ago   Modified 8 years, 3 months ago

Viewed 916 times



4



On our web application, the search results are displayed in sortable tables. The user can click on any column and sort the result. The problem is some times, the user does a broad search and gets a lot of data returned. To make the sortable part work, you probably need all the results, which takes a long time. Or I can retrieve few results at a time, but then sorting won't really work well. What's the best practice to display sortable tables that might contain lots of data?

Thanks for all the advises. I will certainly going over these.

We are using an existing Javascript framework that has the sortable table; "lots" of results means hundreds. The problem is that our users are at some remote site and a lot of delay is the network time to send/receive data from the data center. Sorting the data at the database side and only send one page worth of results at a time is nice; but when the user clicks some column header, another round trip is done, which always add 3-4 seconds.

Well, I guess that might be the network team's problem :)

[javascript](#)[database](#)[jquery-ui-sortable](#)[Share](#)[Improve this question](#)[Follow](#)

edited Sep 13, 2016 at 11:36

[Brian Tompsett - 汤莱恩](#)

5,875 ● 72 ● 61 ● 133

asked Sep 22, 2008 at 20:52

[Jon](#)

263 ● 4 ● 10

Could you provide some details as to what type of database you are running and what webstack your using?

– [Aaron Fischer](#) Sep 22, 2008 at 21:10

6 Answers

Sorted by:

Highest score (default)



3

Using sorting paging at the database level is the correct answer. If your query returns 1000 rows, but you're only going to show the user 10 of them, there is no need for the other 990 to be sent across the network.



Here is a mysql example. Say you need 10 rows, 21-30, from the 'people' table:



```
SELECT * FROM people LIMIT 21, 10
```



answered Sep 22, 2008 at 21:04

Share Improve this answer

Follow



tyshock

1,311 ● 8 ● 13



3

You should be doing paging back on the database server. E.g. on SQL 2005 and SQL 2008 there are paging techniques. I'd suggest looking at paging options for whatever system you're looking at.



Share Improve this answer

answered Sep 22, 2008 at 20:54



Follow



Orion Adrian

19.5k ● 13 ● 54 ● 67



2

What database are you using as there some good paging option in SQL 2005 and upwards using ROW\_NUMBER to allow you to do paging on the server. I found this good one on [Christian Darie's blog](#)



eg This procedure which is used to page products in a category. You just pass in the pagenumber you want and the number of products on the page etc



```
CREATE PROCEDURE GetProductsInCategory
(@CategoryID INT,
@DescriptionLength INT,
@PageNumber INT,
@ProductsPerPage INT,
@HowManyProducts INT OUTPUT)
AS

-- declare a new TABLE variable
DECLARE @Products TABLE
(RowNumber INT,
ProductID INT,
```

```

Name VARCHAR(50),
Description VARCHAR(5000),
Price MONEY,
Image1FileName VARCHAR(50),
Image2FileName VARCHAR(50),
OnDepartmentPromotion BIT,
OnCatalogPromotion BIT)

-- populate the table variable with the complete list
INSERT INTO @Products
SELECT ROW_NUMBER() OVER (ORDER BY Product.ProductID),
       Product.ProductID, Name,
       SUBSTRING(Description, 1, @DescriptionLength) +
       Price, Image1FileName, Image2FileName, OnDepart
OnCatalogPromotion
FROM Product INNER JOIN ProductCategory
  ON Product.ProductID = ProductCategory.ProductID
WHERE ProductCategory.CategoryID = @CategoryID

-- return the total number of products using an OUTPUT
SELECT @HowManyProducts = COUNT(ProductID) FROM @Produ

-- extract the requested page of products
SELECT ProductID, Name, Description, Price, Image1File
       Image2FileName, OnDepartmentPromotion, OnCatalo
FROM @Products
WHERE RowNumber > (@PageNumber - 1) * @ProductsPerPage
       AND RowNumber <= @PageNumber * @ProductsPerPage

```

Share Improve this answer

answered Sep 23, 2008 at 1:44

Follow



Leo Moore

2,158 ● 2 ● 19 ● 22



0

You could do the sorting on the server. AJAX would eliminate the necessity of a full refresh, but there'd still be a delay. Sides, databases a generally very fast at sorting.



Share Improve this answer

answered Sep 22, 2008 at 20:56

Follow



sblundy

61.4k ● 22 ● 123 ● 124



0

For these situations I employ techniques on the SQL Server side that not only leverage the database for the sorting, but also use custom paging to ONLY return the specific records needed.



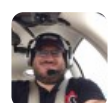
It is a bit of a pain to implemement at first, but the performance is amazing afterwards!



Share Improve this answer

answered Sep 22, 2008 at 20:58

Follow



Mitchel Sellers

63.1k ● 15 ● 114 ● 174



0

How large is "a lot" of data? Hundreds of rows? Thousands?



Sorting can be done via JavaScript painlessly with [Mochikit Sortable Tables](#). However, if the data takes a long time to sort (most likely a second or two [or three!]) then you may want to give the user some visual cue that soming is happening and the page didn't just freeze. For





example, tint the screen (a la Lightbox) and display a "sorting" animation or text.

Share Improve this answer

answered Sep 22, 2008 at 21:01

Follow



**Zack The Human**

**8,481** ● 8 ● 42 ● 61

---