Is a view faster than a simple query?

Asked 10 years, 8 months ago Active 10 months ago Viewed 197k times

325

107

Is a

```
select *  from myView
```

faster than the query itself to create the view (in order to have the same resultSet):

```
select * from ([query to create same resultSet as myView])
```

?

It's not totally clear to me if the view uses some sort of caching making it faster compared to a simple query.

sql sql-server performance

edited Aug 27 '12 at 18:13

Peter Mortensen
14.4k 19 88 117

asked Jan 13 '09 at 14:09

JohnIdol
30.7k 58 147 229

6 Iam not sure about one view, but nested views is total performance hell. – Muflix Feb 12 '17 at 17:03

14 Answers

Yes, views can have a clustered index assigned and, when they do, they'll store temporary results that can speed up resulting queries.

First, simple views are expanded in place and so do not directly contribute to performance improvements - that much is true. **However,** indexed views can *dramatically* improve performance.

Let me go directly to the documentation:

> After a unique clustered index is created on the view, the view's result set is materialized immediately and persisted in physical storage in the database, saving the overhead of performing this costly operation at execution time.

Second, these indexed views can work *even when they are not directly referenced by another query* as the optimizer will use them in place of a table reference when appropriate.

Again, the documentation:

> The indexed view can be used in a query execution in two ways. The query can reference the indexed view directly, or, more importantly, the query optimizer can select the view if it determines that the view can be substituted for some or all of the query in the lowest-cost query plan. In the second case, the indexed view is used instead of the underlying tables and their ordinary indexes. The view does not need to be referenced in the query for the query optimizer to use it during query execution. This allows existing applications to benefit from the newly created indexed views without changing those applications.

This documentation, as well as charts demonstrating performance improvements, can be found here.

Update 2: the answer has been criticized on the basis that it is the "index" that provides the performance advantage, not the "View." However, this is easily refuted.

Let us say that we are a software company in a small country; I'll use Lithuania as an example. We sell software worldwide and keep our records in a SQL Server database. We're very successful and so, in a few years, we have 1,000,000+ records. However, we often need to report sales for tax purposes and we find that we've only sold 100 copies of our software in our home country. By creating an indexed view of just the Lithuanian records, we get to keep the records we need in an indexed cache as described in the MS documentation. When we run our reports for Lithuanian sales in 2008, our query will search through an index with a depth of just 7 (Log2(100) with some unused leaves). If we were to do the same without the VIEW and just relying on an index into the table, we'd have to traverse an index tree with a search depth of 21!

Clearly, the View itself would provide us with a performance advantage (3x) over the simple use of the index alone. I've tried to use a real-world example but you'll note that a simple list of Lithuanian sales would give us an even greater advantage.

Update 3: The question has come up about whether an Indexed View just uses an index placed on the underlying table. That is, to paraphrase: "an indexed view is just the equivalent of a standard index and it offers nothing new or unique to a view." If this was true, of course, then the above analysis would be incorrect! Let me provide a quote from the Microsoft documentation that demonstrate why I think this criticism is not valid or true:

> Using indexes to improve query performance is not a new concept; however, indexed views provide additional performance benefits that cannot be achieved using standard indexes.

Together with the above quote regarding the persistence of data in physical storage and other information in the documentation about how indices are created on Views, I think it is safe to say that an Indexed View is **not** just a cached SQL Select that happens to use an index defined on the main table. Thus, I continue to stand by this answer.

edited Nov 15 '09 at 12:47

community wiki
8 revs, 2 users 88%
Mark Brittingham

24 Yes, indexed views can dramatically improve performance. But indexed views are not just "views", and generally speaking, normal "view" aren't faster than their associated queries. – BradC Jan 13 '09 at 16:17

8 @Charles - it doesn't matter if it's the index, the fact that a view can leverage the index and a raw query can't is enough – annakata Jan 13 '09 at 17:36

171 /applaud @Mark for standing his ground and rationally arguing this one out – annakata Jan 13 '09 at 17:38

15 Oh, man, I've gotten 8 downvotes on this one! I'm amazed that people would downvote so quickly without at least having Charles' courage to argue their point. – Mark Brittingham Jan 13 '09 at 18:45

6 Since a table can only have one clusterdee index, and you CAN create a separate clustered index on a view, (since the fields in the clustered index are independently persisted in the index pages), this is a cheat (work-arounnd?) that allows you to get TWO clustered indices on one Table. – Charles Bretana Jan 13 '09 at 20:34

41

Generally speaking, no. Views are primarily used for convenience and security, not for speed improvements.

That said, SQL Server 2000 and above do have a special feature called **Indexed Views** that *can* greatly improve performance, but you have to create indexed views following a very specific set of guidelines

Here is an article that describes the benefits and creation of indexed views:

> For many years, Microsoft® SQL Server™ has supported the ability to create virtual tables known as views. Historically, these views served these main purposes:
>
> - To provide a security mechanism that restricts users to a certain subset of data in one or more base tables.
> - To provide a mechanism that allows developers to customize how users can logically view the data stored in base tables.
>
> With SQL Server 2000, the functionality of SQL Server views was expanded to provide system performance benefits. It is possible to create a unique clustered index on a view, as well as nonclustered indexes, to improve data access performance on the most complex queries. In SQL Server 2000 and 2005, a view that has a unique clustered index is referred to as an indexed view.

edited Nov 15 '09 at 22:20

Peter Mortensen
14.4k 19 88 117

answered Jan 13 '09 at 16:27

BradC
34.5k 12 61 89

totally disagree... reading from a view allows the SQL to be rewritten.. and it's generally FASTER to read from a view (than from a dump of the view). – Aaron Kempf Jan 9 '13 at 20:23

@AaronKempf, I'd love to see some reference on that, that hasn't been my experience. When I search for "view SQL rewritten", all the results I get refer to Oracle, not SQL server, e.g. docs.oracle.com/cd/E14072_01/server.112/e10810/qrbasic.htm – BradC Jan 9 '13 at 21:18

I was just doing some benchmarking on it yesterday, I was stunned.. basically if I take a dump from a view (into a table) any query that I run is SLOWER.. because most queries pass through the view like butter and get rewritten by the query optimizer.. At least that's what I assume. I'll try to write a blog entry on it soon, the benchmarking was quite fascinating stuff.. Basically views help performance tremendously. – Aaron Kempf Jan 10 '13 at 20:58

@AaronKempf Not sure that's even the same scenario as the original question (which is about a query vs putting that identical query in a view). Anyway, I can't see how materializing a view into a table would make it SLOWER (that's exactly what an indexed view does), unless your new table doesn't have good indexes. – BradC Jan 10 '13 at 21:37

1 Brad; I wrote a really bad blog post about how views are saving me 99% of my performance in this situation.. I'm planning on writing a couple of other articles, but I know that I need to put a TON more detail into this.. Do you mind taking a look at this and telling me what you think of my description? I know it won't make more sense (until I get 2-3 other articles up).. but I'm madly in love with views, and I have been for a long time! accessadp.com/2013/01/22/do-views-increase-performance – Aaron Kempf Jan 22 '13 at 14:59

In SQL Server at least, Query plans are stored in the plan cache for both views and ordinary SQL queries, based on query/view

So no, there is no difference, given that you are reusing the same SQL query and the same view with the same frequency.

Obviously, in general, a view, by it's very nature (That someone thought it was to be used often enough to make it into a view) is generally more likely to be "reused" than any arbitrary SQL statement.

answered Jan 13 '09 at 14:21

Charles Bretana
118k 20 129 201

13

EDIT: I was wrong, and you should see Marks answer above.

I cannot speak from experience with SQL Server, but for most databases the answer would be no. The only potential benefit that you get, performance wise, from using a view is that it could potentially create some access paths based on the query. But the main reason to use a view is to simplify a query or to standardize a way of accessing some data in a table. Generally speaking, you won't get a performance benefit. I may be wrong, though.

I would come up with a moderately more complicated example and time it yourself to see.

edited May 23 '17 at 11:47

Community ♦
1 1

answered Jan 13 '09 at 14:11

Ryan Guill
9,739 4 31 47

1 another reason for views is to assist access control in role based models – annakata Jan 13 '09 at 14:32

1 You are wrong about the performance improvements. I did not explain enough to convince some people in my original comment but MS has explicit documentation on how to use views to improve performance. See my (now heavily downvoted) response below. – Mark Brittingham Jan 13 '09 at 14:38

5

It may be faster if you create a materialized view (***with schema binding***). Non-materialized views execute just like the regular query.

edited Aug 27 '12 at 18:17

Peter Mortensen
14.4k 19 88 117

answered Jan 13 '09 at 14:12

Otávio Décio
63.9k 14 147 215

4

My understanding is that a while back, a view would be faster because SQL Server could store an execution plan and then just use it instead of trying to figure one out on the fly. I think the performance gains nowadays is probably not as great as it once was, but I would have to guess there would be some marginal improvement to use the view.

edited Aug 27 '12 at 18:18

Peter Mortensen
14.4k 19 88 117

answered Jan 13 '09 at 14:12

E.J. Brennan
35.8k 4 62 100

This was my understanding: used to matter, doesn't any more – annakata Jan 13 '09 at 14:33

Doesn't from a performance standpoint - does as a means of access restriction. But that would be another topic. ;) – AnonJr Jan 13 '09 at 14:46

oh sure, *lots* of good reasons to be using views, not a single one to use raw queries :P – annakata Jan 13 '09 at 17:37

2

Definitely a view is better than a nested query for SQL Server. Without knowing exactly why it is better (until I read Mark Brittingham's post), I had run some tests and experienced almost shocking performance improvements when using a view versus a nested query. After running each version of the query several hundred times in a row, the view version of the query completed in half the time. I'd say that's proof enough for me.

answered Jan 13 '09 at 16:03

Jordan

Thanks Jordan...glad to hear that all this theory works out in the *real* world. – Mark Brittingham Jan 13 '09 at 18:42

i have experience with nested view (view in view) and there was very bad performance. When all views was rewrited to sub selects, performance was many times faster, so maybe there is a place to some serious testing. – Muflix Feb 12 '17 at 17:10

1

I would expect the two queries to perform identically. A view is nothing more than a stored query definition, there is no caching or storing of data for a view. The optimiser will effectively turn your first query into your second query when you run it.

If the view is a small set of fields and those fields are then covered with an index is SQL Server clever enough to use that covering index when fulfilling the second form of query? – AnthonyWJones Jan 13 '09 at 14:18

1

It all depends on the situation. MS SQL Indexed views are faster than a normal view or query but indexed views can not be used in a mirrored database invironment (MS SQL).

A view in any kind of a loop will cause serious slowdown because the view is repopulated each time it is called in the loop. Same as a query. In this situation a temporary table using # or @ to hold your data to loop through is faster than a view or a query.

So it all depends on the situation.

answered Mar 15 '13 at 14:24

Dasboot
11 1

0

There is no practical different and if you read BOL you will find that ever your plain old SQL SELECT * FROM X does take advantage of plan caching etc.

answered Jan 13 '09 at 14:15

keithwarren7
9,827 6 44 71

0

There should be some trivial gain in having the execution plan stored, but it will be negligible.

answered Jan 13 '09 at 14:31

JosephStyons
37.9k 57 147 224

0

the query is truly very simple, the benefits may be negligible.

Now, if you're doing a complex query, create the view.

answered Jan 13 '09 at 14:35
CAReed

0

In my finding, using the view is a little bit faster than a normal query. My stored procedure was taking around 25 minutes (working with a different larger record sets and multiple joins) and after using the view (non-clustered), the performance was just a little bit faster but not significant at all. I had to use some other query optimization techniques/method to make it a dramatic change.

edited Aug 27 '12 at 18:22

Peter Mortensen
14.4k 19 88 117

answered Oct 7 '11 at 9:18

kta
13.9k 6 51 42

How we are writing/designing the query is very important. – kta Oct 7 '11 at 9:28

I have found using CTE's to limit the data returning from a table and then doing all the work/joins, etc off the CTE's has dramatically improved performance in many cases – MattE May 15 at 17:02

0

Select from a View or from a table will not make too much sense.

Of course if the View does not have unnecessary joins, fields, etc. You can check the execution plan of your queries, joins and indexes used to improve the View performance.

You can even create index on views for faster search requirements. http://technet.microsoft.com/en-us/library/cc917715.aspx

But if you are searching like '%...%' than the sql engine will not benefit from an index on text column. If you can force your users to make searches like '...%' than that will be fast

referred to answer on asp forums : https://forums.asp.net/t/1697933.aspx?Which+is+faster+when+using+SELECT+query+VIEW+or+Table+