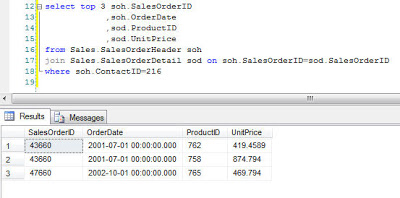
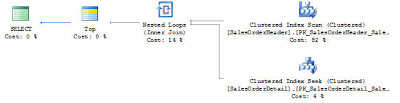
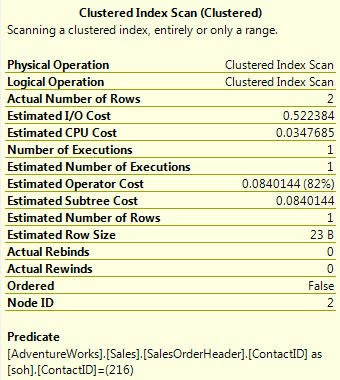
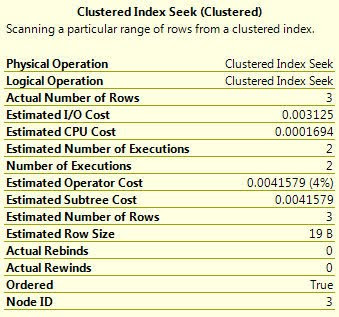
**Friday, November 19, 2010**

**A Second in the Life of a Query Operator**

**FROM:** [**http://bradsruminations.blogspot.com/2010/11/second-in-life-of-query-operator.html?utm\_source=Brent+Ozar+Unlimited%C2%AE+List&utm\_campaign=e9491fcb77-Weekly\_Links\_2017\_04\_09&utm\_medium=email&utm\_term=0\_8e3e861dd9-e9491fcb77-296071565&goal=0\_8e3e861dd9-e9491fcb77-296071565&mc\_cid=e9491fcb77&mc\_eid=b137e91a2b**](http://bradsruminations.blogspot.com/2010/11/second-in-life-of-query-operator.html?utm_source=Brent+Ozar+Unlimited%C2%AE+List&utm_campaign=e9491fcb77-Weekly_Links_2017_04_09&utm_medium=email&utm_term=0_8e3e861dd9-e9491fcb77-296071565&goal=0_8e3e861dd9-e9491fcb77-296071565&mc_cid=e9491fcb77&mc_eid=b137e91a2b)

There are countless books and blogs out there that tell you that you should read a query plan from right to left.  
  
But unless you’re used to reading Hebrew, isn’t that backwards? We read everything else from left to right… Why should we read query plans in the opposite direction?  
  
The answer is: We really shouldn’t. You’ve been misinformed.  
  
Yes, it’s true that the flow of information is from right to left… All those arrows that are pointing in that direction indicate that. We can see the sources of the data on the right and the eventual destination on the left.  
  
But the query does not *start* with the operators on the right.   
  
Let’s look at an example:  
  
[](http://1.bp.blogspot.com/_vPa0QtSASSk/TOcmdfNGBfI/AAAAAAAAAdM/HJzWudigld4/s1600/Top3Query.jpg)  
  
This is a simple query that produces 3 rows of data. Here is the (actual) execution plan:  
  
[](http://3.bp.blogspot.com/_vPa0QtSASSk/TOcmriYL-nI/AAAAAAAAAdU/oV1bl1OFQGU/s1600/Top3Plan.jpg)  
  
The arrows are pointing from right to left, making it look like the Clustered Index Scan and Clustered Index Seek are the ones in charge. But they are NOT. In reality, they are just dumb file cabinets of data that only do what they are told.  
  
Take a look at the properties of the Clustered Index Scan operator:  
  
[](http://1.bp.blogspot.com/_vPa0QtSASSk/TOcm2V3KxdI/AAAAAAAAAdc/hKSEWKz6VEA/s1600/ClustScanProps.jpg)  
  
The job of a Scan operator is to scan the entire clustered index from beginning to end, isn’t it? In this case, it’s supposed to apply the predicate and only pay attention to those rows with ContactID 216. But the Actual Rows in the property sheet is equal to only 2. There are a heckuva lot more than 2 rows with ContactID 216 in the table. How did this operator know when to stop? Isn’t the TOP operator in charge of limiting the rows? But he’s two positions away from the Scan Operator in the plan!   
  
And look at the properties of the Clustered Index Seek:  
  
[](http://1.bp.blogspot.com/_vPa0QtSASSk/TOcm9Sxe64I/AAAAAAAAAdk/mk4PsDUR6GM/s1600/ClustSeekProps.jpg)  
  
It shows that the Number of Executions is equal to 2. Why does it execute twice? And what does that have to do with 3 rows?  
  
The actual execution of a query does not consist of the operators on the right *pushing* their rows to the left. Instead, query execution consists of operators on the left *requesting* rows from their child operators to their right.  
  
Just before you start a query, all those operators are asleep. Pushing the *Execute* button in SSMS wakes up the operator on the far left (the SELECT in our example above), and that operator is in charge of waking up the other operators down the tree and instigating the action of acquiring and outputting rows.  
  
Let’s take a closer, more intimate look at what *really* goes on in this plan when you hit the *Execute* button. We’ll do it in the form of a dramatic play, just to make it more interesting and up close and personal.  
  
So sit back and enjoy. (Note: For those of you reading this on your phone or other mobile device or certain readers, you may not be able to enjoy the full experience. The entire script below is presented in resplendent color for your convenience and amusement).

**Brad Schulz Productions Proudly Presents...  
  
*A Second in the Life of a Query Operator*  
  
A Play in One Act**

***Dramatis Personæ:***  
  
**SQL Engine:** It is assumed that you already know this character quite well.  
  
**Simon Select:** A very efficient, reliable man with impeccable manners. Imagine the butler [Hudson in *Upstairs, Downstairs*](http://en.wikipedia.org/wiki/Gordon_Jackson_(actor)).  
  
**Tony Top:** A somewhat impatient guy with a tough exterior that masks a sometimes kinder man inside. Imagine [Tony Soprano](http://en.wikipedia.org/wiki/Tony_soprano).  
  
**Nora Nested-Loops:** A no-nonsense woman who does her work without complaint. Imagine [Mrs. Doubtfire](http://en.wikipedia.org/wiki/Mrs_Doubtfire).  
  
**Stuart Scan:** A simple-minded country boy who is eager to please. Imagine [Gomer Pyle](http://www.imdb.com/title/tt0057752/).  
  
**Stanley Seek:** An uncomplicated man who’s very fond of napping, since he never seems to get enough sleep. Imagine [Dagwood Bumstead](http://en.wikipedia.org/wiki/Dagwood_Bumstead).

**Act I Scene 1.**  
  
The Setting: A Query Plan Universe, consisting of 5 rooms, each containing an Operator, all asleep in their beds.  
  
**VOICE OF SQL ENGINE:** Simon, wake up. I have a job for you.  
  
Simon wakes up and promptly gets up from his bed, already wearing a suit with a black swallowtail coat that, amazingly, has no wrinkles in it whatsoever. He straightens his tie.  
  
**SIMON:** Very good, sir. I will attend to it immediately.   
  
He posts a sign in the status bar that reads *“Executing query…”* and he places a little green spinner icon next to it to indicate that the SPID is busy.   
  
[Executing Query...](http://1.bp.blogspot.com/_vPa0QtSASSk/TOcnFC1ftMI/AAAAAAAAAds/rX0gJ59sqOM/s1600/ExecutingQuery.jpg)  
  
Then he walks to a door on the opposite side of his room. He has no idea who is in the room behind the door. He politely knocks.   
  
**SIMON:** (Opening the door a crack) Excuse me.   
  
**TONY:** Huh?   
  
Tony rubs his eyes and sits up at the edge of his bed.   
  
**SIMON:** Oh, good morning, Mister Top.   
  
**TONY:** (Yawning) I keep tellin’ ya, Simon, you don’t have to call me Mister Top. You can call me Tony. How’re ya doin’?   
  
**SIMON:** Just fine, thank you, Mis---… er… Tony. I have been engaged to display some rows in the SSMS Results Window, and I require your assistance.   
  
**TONY:** Yeah, sure, no problem. I know the drill.   
  
**SIMON:** Let us proceed, then. May I please have a row of data?   
  
**TONY:** Comin’ right up. Just give me a millisecond.   
  
Tony walks across to the door on the opposite side of his room and throws it open, banging the door against the wall.   
  
**TONY:** Hey you! Get your a\*\* outta bed! I got a job for ya.   
  
Nora is startled out of her sleep by the loud voice, but she soon regains composure and gets up from her bed and approaches Tony and points her finger at him.   
  
**NORA:** Mister Top! How dare you barge in like that! And there’s no need to be so coarse. I know that I have a job to do, but I will not stand for vulgarity, is that clear?   
  
**TONY:** Yeah, yeah, okay, sorry, Nora. I didn’t know it was you. Many times the query operators I work with are pretty slow in responding. You ever work with Sonny Sort? Jeez, he takes forever!   
  
**NORA:** I understand that, but one should always try to treat his or her coworkers with respect.   
  
**TONY:** Yeah, sure, Nora, whatever. Well, let’s get to work. Gimme a --- … Sorry. (He clears his throat) May I please have a row of data?   
  
**NORA:** That’s much better, Tony, dear. I’m glad we’re getting down to business. I’ll have that row for you in no time.   
  
Nora walks to the opposite side of her room, where there are two doors. She opens the door on the left.   
  
**NORA:** Rise and shine! We’ve got work to do.   
  
Stuart stirs in his bed and looks toward the door.   
  
**STUART:** Well, gawwwllleeee! If it ain’t Nora Nested-Loops!   
  
**NORA:** Hello, Stuart.   
  
**STUART:** Well I’m about as excited as a bag full of puppies to be working with you again, Nora. How long has it been since we worked together?   
  
**NORA:** About 12 seconds, dear.   
  
**STUART:** Hooo-wheee! Has it really been that long? Well, what can I do for you?   
  
**NORA:** You know the routine, Stuart. I need a row.   
  
**STUART:** Right away.   
  
Stuart walks over the file cabinet in his room and looks at the work order that’s tacked to the wall above it.   
  
[Stuart Scan's Predicate](http://4.bp.blogspot.com/_vPa0QtSASSk/TOcnOtBwoZI/AAAAAAAAAd0/CGtWQvM7UFk/s1600/ScanPredicate.jpg)  
  
**STUART:** Looks like I have to get rows just for ContactID 216 from this here clustered index. I sure do hope that this clustered index ain’t too large. A second or two ago, I had a job where I had to scan over 10,000,000 rows, and, boy, I was busier than a long-tailed cat in a room full of rocking chairs! In fact, I said to myself that---  
  
**NORA:** Time waits for no one, Stuart dear. I need more action and less talk.   
  
**STUART:** Oh, sure, sorry, Nora.   
  
He opens the drawer of the file cabinet. It’s full of pages of data, with each page consisting of rows in order of SalesOrderID. He takes out the first page and examines each row on the page, looking for ones with a ContactID value of 216. He finds one. He checks the work order once again.   
  
[Stuart Scan's Output List](http://2.bp.blogspot.com/_vPa0QtSASSk/TOcnYg2WndI/AAAAAAAAAd8/4mJVdpYGw5g/s1600/ScanOutputList.jpg)  
  
**STUART:** Looks like I only gotta give you the SalesOrderID and OrderDate columns.   
  
He pulls those columns from the row and walks back to the door to hand them (as a row) to Nora.   
  
**STUART:** Here ya go, Nora.   
  
**NORA:** Thank you, Stuart.   
  
Nora looks at her work order and sees that she is to perform an INNER JOIN using an outer reference of SalesOrderID. She examines the row in her left hand that she received from Stuart and then opens the second door, which is to the right of the door to Stuart’s room.   
  
**NORA:** Time to get up!   
  
Stanley is startled out of his sleep and falls out of his bed and bumps his head.   
  
**STANLEY:** (Rubbing the back of his head) Time to get up already? It seems like I’ve only been sleeping for a few hundred milliseconds.   
  
**NORA:** Yes, it’s that time again, Stanley. We’ve got a job to do. (She looks again at the row she received from Stuart in her left hand) I need you to get me a row for SalesOrderID 43660.   
  
**STANLEY:** (Yawning) Okay, just a millisecond.   
  
He walks over to a file cabinet in his room. It has three drawers. They are labeled *43659 thru 51821*, *51822 thru 62790*, and *62791 thru 75123* respectively. Stanley opens the top drawer, and looks through the folders inside and immediately finds one that is labeled *43659 thru 43875*. He leafs through the pages inside and immediately finds the page containing rows for SalesOrderID 43660. He checks his work order.   
  
[Stanley Seek's Output List](http://2.bp.blogspot.com/_vPa0QtSASSk/TOcngKj4iZI/AAAAAAAAAeE/9m1RLZROiGU/s1600/SeekOutputList.jpg)  
  
**STANLEY:** Looks like I’m only supposed to give you the ProductID and UnitPrice columns.   
  
He pulls those columns from the row and walks back to the door to hand them (as a row) to Nora.   
  
**STANLEY:** Here it is, Nora… A row for SalesOrderID 43660.   
  
**NORA:** Thank you, dear.   
  
Nora accepts the row in her right hand. She walks over to the copy machine in her room and makes a Xerox copy of the row she has in her left hand (from Stuart). She takes the copy of that row and joins it with the row in her right hand (from Stanley), creating a new row consisting of columns SalesOrderID, OrderDate, ProductID, and UnitPrice. She walks to the door where Tony is waiting.   
  
**NORA:** Here’s your row, Tony.   
  
**TONY:** Thanks, Nora.   
  
Tony accepts the row and looks over his work order.   
  
[Tony Top's Expression](http://4.bp.blogspot.com/_vPa0QtSASSk/TOcnzb7RuxI/AAAAAAAAAeM/VMaf6LiwY0o/s1600/TopExpression.jpg)  
  
**TONY:** (to himself) Only 3 rows, huh? Okay. (He makes a tick mark to keep track of the number of rows he receives) That’s number one.   
  
He walks over to the door where Simon is waiting.   
  
**TONY:** Here ya go, Simon.   
  
**SIMON:** Thank you, Mister Top.   
  
Simon accepts the row. He meticulously writes the columns of the row into a nice grid structure in the SSMS Results Window, using the font indicated in his work order. Then he walks back to the door.   
  
**SIMON:** Another row, please, Mister Top.   
  
**TONY:** Okay, I’m on it. (Walking to the opposite door) Nora, I need another row.   
  
**NORA:** Alright, Tony. (She looks at the row from Stuart that’s still in her left hand and walks to Stanley’s door) Stanley, dear, I need another row for SalesOrderID 43660.   
  
**STANLEY:** (Jerking his head up from almost nodding off) Huh? Oh, okay, Nora.   
  
He goes back to the page where he had acquired the previous row. He finds another row for SalesOrderID 43660 and pulls out the appropriate columns and walks back towards Nora.   
  
**STANLEY:** Here you go, Nora.   
  
**NORA:** Thank you.   
  
Nora once again makes a Xerox copy of the row in her left hand and joins that copy with the row acquired from Stanley and walks back to Tony.   
  
**NORA:** Here you are, Tony dear.   
  
**TONY:** Thanks. (Making another tick mark on his worksheet) That’s two. (Walking back to Simon) Here’s your row, Simon.   
  
**SIMON:** Very good, Mister Top.   
  
Simon writes the columns of the new row to the grid in the SSMS Results Window and returns once again to Tony’s door.   
  
**SIMON:** Another row, if you please, Mister Top.   
  
**TONY:** Okay, you got it. (Walking to the opposite door) Nora, I need another row.   
  
**NORA:** Alright, Tony. (She looks once again at the row from Stuart that’s still in her left hand and walks to Stanley’s door) Stanley, dear, once again I need another row for SalesOrderID 43660.   
  
**STANLEY:** (Yawning) Okay.   
  
He goes back to the page where he had acquired the previous row. But there isn’t another row for SalesOrderID 43660.   
  
**STANLEY:** Sorry, Nora, I’ve already given you all I’ve got for that SalesOrderID.   
  
**NORA:** Well, thank you for your help, Stanley. You can go back to sleep now.   
  
**STANLEY:** Thank goodness!   
  
As Stanley shuts the drawer of the file cabinet and rushes towards the bed, Nora closes his door. She throws the row in her left hand into the trash and then goes to the door on her left, where Stuart is standing.   
  
**NORA:** Stuart, dear, I need another row from you.   
  
**STUART:** Okay, Nora… I can get that for you faster than green grass goes through a goose.   
  
He goes to the file cabinet and continues scanning rows, looking for the next one with ContactID 216.   
  
**STUART:** I found one. (Walking back towards Nora) Here it is.   
  
**NORA:** Thank you, dear.   
  
She accepts the row in her left hand and examines the SalesOrderID in the row, then walks over to Stanley’s door and opens it.   
  
**NORA:** Wake up, Stanley, dear! I need a row for SalesOrderID 47660.   
  
Stanley falls out of his bed, bumping his head once again.   
  
**STANLEY:** (Groaning) What? You told me to go back to sleep only a few nanoseconds ago.   
  
**NORA:** Well, that’s the nature of the job, Stanley, dear. Every time I get a row from Stuart here, I have to wake you up and get rows from you to join to his. Once you give me all the rows relating to Stuart’s then you get to go back to sleep. This is my second row from Stuart, so I’m waking you up for the second time. Every time I wake you up, it’s called an execution.  
  
**STANLEY:** Yeah, I know. But all this going to sleep and waking up over and over again is killing me! I think the word “execution” is very appropriate! (Yawning) Okay, hold on… What was that SalesOrderID again?   
  
**NORA:** 47660, dear.   
  
**STANLEY:** Okay.   
  
Stanley goes to his file cabinet and opens the appropriate drawer and the appropriate folder and the appropriate page and finds the first row for SalesOrderID 47660.   
  
**STANLEY:** Got one. (Walking back to Nora) Here you go.   
  
**NORA:** Thank you.   
  
Nora once again makes a Xerox copy of the row in her left hand and joins that copy with the row acquired from Stanley and walks back to Tony.   
  
**NORA:** Here you are, Tony dear.   
  
**TONY:** Thanks. (Making another tick mark on his worksheet) That’s three. (Walking back to Simon) Here ya go, Simon.   
  
**SIMON:** Excellent… Thank you, Mister Top.   
  
Simon adds the new row to the grid in the SSMS Results Window and returns once again to Tony’s door.   
  
**SIMON:** May I have another row, please, Mister Top?   
  
**TONY:** (Double-checking his tick marks against his work order) Sorry, Simon, but I can’t give you any more rows. I’ve reached my quota.   
  
**SIMON:** Well, I suppose that’s it, then. Thank you for your assistance, Mister Top. Time to shut down the operation.   
  
**TONY:** Okay, Simon. Hey, Nora, time to shut down.   
  
**NORA:** Alrighty. Stanley and Stuart, it’s time to shut down. You can go both go back to sleep.   
  
**STANLEY:** Yes! Sleep-time again!   
  
Stanley closes his door and jumps into his bed and is out cold before his head even hits the pillow.   
  
**STUART:** Okay, Nora. I actually am a little tired. In fact, I’m more exhausted than a mule in---  
  
Nora closes his door before he can finish his hokey down-home analogy. Stuart shrugs his shoulders and climbs into his bed and goes to sleep.   
  
**NORA:** (Walking back to the door of Tony’s room) Good night, Tony, dear.   
  
**TONY:** See ya later, Nora.   
  
Nora closes the door, turns off the light, and slips into her bed.   
  
**TONY:** (Walking back to the door of Simon’s room) Well, Simon, until next time…  
  
**SIMON:** Always a pleasure, Mister Top.   
  
Tony closes the door, turns off the light, and gets into his bed.   
  
Simon writes the message *“(3 row(s) affected)”* to the Messages Window. Then he posts a sign that reads *“Query executed successfully.”* and removes the green spinner icon, replacing it with a checkmark icon.   
  
[Another Job Well Done](http://4.bp.blogspot.com/_vPa0QtSASSk/TOcn7HkaYgI/AAAAAAAAAeU/hB5zeF_XuHY/s1600/QueryExecutedSuccessfully.jpg)  
  
Then Simon sits at the edge of his bed and smiles to himself on another job well done. He lies down and reaches over for the light and turns it off.  
  
And then he gently closes his eyes.   
  
**THE END.**

Posted by [Brad Schulz](https://www.blogger.com/profile/01852762873611487368) at [6:10 PM](http://bradsruminations.blogspot.com/2010/11/second-in-life-of-query-operator.html) [http://img1.blogblog.com/img/icon18_email.gif](https://www.blogger.com/email-post.g?blogID=3000721125799986885&postID=5317702214432572923)[https://resources.blogblog.com/img/icon18_edit_allbkg.gif](https://www.blogger.com/post-edit.g?blogID=3000721125799986885&postID=5317702214432572923&from=pencil)

[Email This](https://www.blogger.com/share-post.g?blogID=3000721125799986885&postID=5317702214432572923&target=email)[BlogThis!](https://www.blogger.com/share-post.g?blogID=3000721125799986885&postID=5317702214432572923&target=blog)[Share to Twitter](https://www.blogger.com/share-post.g?blogID=3000721125799986885&postID=5317702214432572923&target=twitter)[Share to Facebook](https://www.blogger.com/share-post.g?blogID=3000721125799986885&postID=5317702214432572923&target=facebook)[Share to Pinterest](https://www.blogger.com/share-post.g?blogID=3000721125799986885&postID=5317702214432572923&target=pinterest)

Labels: [Query Plan](http://bradsruminations.blogspot.com/search/label/Query%20Plan), [Syndicated](http://bradsruminations.blogspot.com/search/label/Syndicated)

**25 comments:**

http://lh3.googleusercontent.com/zFdxGE77vvD2w5xHy6jkVuElKv-U9_9qLkRYK8OnbDeJPtjSZ82UPq5w6hJ-SA=s35

[**bbacher**](https://www.blogger.com/profile/05796899364523845418)[November 20, 2010 at 2:27 PM](http://bradsruminations.blogspot.com/2010/11/second-in-life-of-query-operator.html?showComment=1290292049367#c2669586373853960861)

That's Entertainment!  
  
But... how can I use this knowledge to improve my queries?

[Reply](javascript:;)[Delete](https://www.blogger.com/delete-comment.g?blogID=3000721125799986885&postID=2669586373853960861)

1. http://2.bp.blogspot.com/_mzdqzg2V9tM/S-Du4L1_VWI/AAAAAAAAAAk/3PVxQtSvsv0/S45-s35/Me%2BRapids.jpg

[**Paul White NZ**](https://www.blogger.com/profile/04690243284528295117)[November 20, 2010 at 11:42 PM](http://bradsruminations.blogspot.com/2010/11/second-in-life-of-query-operator.html?showComment=1290325333116#c8828553651191382747)

Very nice Brad, though I do worry about your sanity sometimes :)  
  
Paul

[Reply](javascript:;)[Delete](https://www.blogger.com/delete-comment.g?blogID=3000721125799986885&postID=8828553651191382747)

1. http://img1.blogblog.com/img/blank.gif

[**Wilfred**](http://www.wilfredvandijk.nl)[November 22, 2010 at 1:53 AM](http://bradsruminations.blogspot.com/2010/11/second-in-life-of-query-operator.html?showComment=1290419606618#c1399162312479412205)

I assume parallel processing has something to do with twins?  
  
Wilfred

[Reply](javascript:;)[Delete](https://www.blogger.com/delete-comment.g?blogID=3000721125799986885&postID=1399162312479412205)

1. http://4.bp.blogspot.com/_dH-WtczbgeY/St3y8ytJ_1I/AAAAAAAAAF8/GoqHaV3FR9Y/S45-s35/415209714_8395830b01_b.jpg

[**Michael J. Swart**](https://www.blogger.com/profile/05408240220683534698)[November 22, 2010 at 7:22 AM](http://bradsruminations.blogspot.com/2010/11/second-in-life-of-query-operator.html?showComment=1290439366229#c4072879469353844602)

That was fun to read. Thanks again Brad

[Reply](javascript:;)[Delete](https://www.blogger.com/delete-comment.g?blogID=3000721125799986885&postID=4072879469353844602)

1. http://1.bp.blogspot.com/_vPa0QtSASSk/SgJcKyrx9-I/AAAAAAAAAAk/mlghz9uMwAI/S45-s35/biggerpicture.jpg

[**Brad Schulz**](https://www.blogger.com/profile/01852762873611487368)[November 22, 2010 at 8:52 AM](http://bradsruminations.blogspot.com/2010/11/second-in-life-of-query-operator.html?showComment=1290444747487#c3730173021307793535)

@bbacher: It won't nbecessarily help you to improve your queries per se, but it does give you more insight into how a query actually works. Once I discovered this, I started looking at queries in a whole new light, and I was able to see what was going on with more complex queries that involve recursion or window functions, for example.

[Reply](javascript:;)[Delete](https://www.blogger.com/delete-comment.g?blogID=3000721125799986885&postID=3730173021307793535)

1. http://1.bp.blogspot.com/_vPa0QtSASSk/SgJcKyrx9-I/AAAAAAAAAAk/mlghz9uMwAI/S45-s35/biggerpicture.jpg

[**Brad Schulz**](https://www.blogger.com/profile/01852762873611487368)[November 22, 2010 at 8:53 AM](http://bradsruminations.blogspot.com/2010/11/second-in-life-of-query-operator.html?showComment=1290444814733#c4534514439948150434)

@Paul: Regarding my sanity: I asked my imaginary friend Harvey about it, and he said that as far as he's concerned, I'm perfectly sane. So there!

[Reply](javascript:;)[Delete](https://www.blogger.com/delete-comment.g?blogID=3000721125799986885&postID=4534514439948150434)

1. http://1.bp.blogspot.com/_vPa0QtSASSk/SgJcKyrx9-I/AAAAAAAAAAk/mlghz9uMwAI/S45-s35/biggerpicture.jpg

[**Brad Schulz**](https://www.blogger.com/profile/01852762873611487368)[November 22, 2010 at 8:54 AM](http://bradsruminations.blogspot.com/2010/11/second-in-life-of-query-operator.html?showComment=1290444865695#c161044623791551587)

@Wilfred: Parallel processing and twins: LOL. Very good! Either that, or Nora's copying machine could be used to create clones.

[Reply](javascript:;)[Delete](https://www.blogger.com/delete-comment.g?blogID=3000721125799986885&postID=161044623791551587)

1. http://img1.blogblog.com/img/blank.gif

**Anonymous**[December 5, 2010 at 10:04 PM](http://bradsruminations.blogspot.com/2010/11/second-in-life-of-query-operator.html?showComment=1291615446778#c6285999316769295319)

Awsome...Thats a very nice and simple way to explain what goes on inside the sql server engine when a query is fired

[Reply](javascript:;)[Delete](https://www.blogger.com/delete-comment.g?blogID=3000721125799986885&postID=6285999316769295319)

1. http://1.bp.blogspot.com/_hrNT9mro9ZA/SgMWRYf2VwI/AAAAAAAAHF0/v6TvIO2vwc4/S45-s35/m3.jpg

[**Aashish**](https://www.blogger.com/profile/00632169663220329519)[December 7, 2010 at 4:47 PM](http://bradsruminations.blogspot.com/2010/11/second-in-life-of-query-operator.html?showComment=1291769250888#c4136885163967966249)

That is simply the BEST way anyone has ever described the workings of a query!  
  
Thanks Brad!  
  
-Aashish

[Reply](javascript:;)[Delete](https://www.blogger.com/delete-comment.g?blogID=3000721125799986885&postID=4136885163967966249)

1. http://1.bp.blogspot.com/_vPa0QtSASSk/SgJcKyrx9-I/AAAAAAAAAAk/mlghz9uMwAI/S45-s35/biggerpicture.jpg

[**Brad Schulz**](https://www.blogger.com/profile/01852762873611487368)[December 7, 2010 at 8:00 PM](http://bradsruminations.blogspot.com/2010/11/second-in-life-of-query-operator.html?showComment=1291780839399#c7036264902459940434)

@Aashish and @Anonymous:  
  
Thanks for the great feedback! I'm so glad you enjoyed it.  
  
--Brad

[Reply](javascript:;)[Delete](https://www.blogger.com/delete-comment.g?blogID=3000721125799986885&postID=7036264902459940434)

1. http://img1.blogblog.com/img/blank.gif

**Anonymous**[December 8, 2010 at 10:10 AM](http://bradsruminations.blogspot.com/2010/11/second-in-life-of-query-operator.html?showComment=1291831828538#c1143657807351187330)

This is just great.Wonderfully written. Very informative. I am sure it took a bit of work on your part to craft a nice tight story and your hard work was not wasted.

[Reply](javascript:;)[Delete](https://www.blogger.com/delete-comment.g?blogID=3000721125799986885&postID=1143657807351187330)

1. http://lh3.googleusercontent.com/zFdxGE77vvD2w5xHy6jkVuElKv-U9_9qLkRYK8OnbDeJPtjSZ82UPq5w6hJ-SA=s35

[**Sabyasachi Mukherjee**](https://www.blogger.com/profile/04459662520963253374)[December 9, 2010 at 10:26 AM](http://bradsruminations.blogspot.com/2010/11/second-in-life-of-query-operator.html?showComment=1291919166819#c7818656312315529706)

Awesome! This is one of the best sqlserver article I ever read. I can literally visualise how the query is being executed .. only downside - I do hesitate to fire a query immediately after another - hoping somewhere Stanley gets some sleep.

[Reply](javascript:;)[Delete](https://www.blogger.com/delete-comment.g?blogID=3000721125799986885&postID=7818656312315529706)

1. http://1.bp.blogspot.com/_vPa0QtSASSk/SgJcKyrx9-I/AAAAAAAAAAk/mlghz9uMwAI/S45-s35/biggerpicture.jpg

[**Brad Schulz**](https://www.blogger.com/profile/01852762873611487368)[December 10, 2010 at 5:25 PM](http://bradsruminations.blogspot.com/2010/11/second-in-life-of-query-operator.html?showComment=1292030707875#c4363504892665163641)

@Sabyasachi:  
  
LOL... poor Stanley!  
  
I have to admit that ever since I wrote this post, I look at query plans in a whole new light, thinking of Nora and Simon and all the rest.  
  
Thank you for your comment!  
  
--Brad

[Reply](javascript:;)[Delete](https://www.blogger.com/delete-comment.g?blogID=3000721125799986885&postID=4363504892665163641)

1. http://img1.blogblog.com/img/blank.gif

[**Michael Washington**](http://openlightgroup.net/Blog/tabid/58/EntryId/141/Silverlight-View-Model-MVVM-A-Play-In-One-Act.aspx)[December 11, 2010 at 3:47 PM](http://bradsruminations.blogspot.com/2010/11/second-in-life-of-query-operator.html?showComment=1292111256772#c7574139097745809438)

I love this blog post. I made an homage to it called "Silverlight View Model (MVVM) - A Play In One Act" http://openlightgroup.net/Blog/tabid/58/EntryId/141/Silverlight-View-Model-MVVM-A-Play-In-One-Act.aspx

[Reply](javascript:;)[Delete](https://www.blogger.com/delete-comment.g?blogID=3000721125799986885&postID=7574139097745809438)

1. http://2.bp.blogspot.com/_mzdqzg2V9tM/S-Du4L1_VWI/AAAAAAAAAAk/3PVxQtSvsv0/S45-s35/Me%2BRapids.jpg

[**Paul White NZ**](https://www.blogger.com/profile/04690243284528295117)[December 15, 2010 at 10:48 AM](http://bradsruminations.blogspot.com/2010/11/second-in-life-of-query-operator.html?showComment=1292438891133#c2120281257579639168)

Brad,  
  
To add to my previous comment:  
  
You did miss a bit from the script concerning the initialization step (no doubt for space reasons). For these details and more, please see http://sqlblog.com/blogs/paul\_white/archive/2010/08/05/iterators-query-plans-and-why-they-run-backwards.aspx  
  
I do still routinely read plans from the top-right and working left, because I am usually interested in the flow of data, rather than the flow of control.  
  
Cheers!  
  
Paul

[Reply](javascript:;)[Delete](https://www.blogger.com/delete-comment.g?blogID=3000721125799986885&postID=2120281257579639168)

1. http://1.bp.blogspot.com/_vPa0QtSASSk/SgJcKyrx9-I/AAAAAAAAAAk/mlghz9uMwAI/S45-s35/biggerpicture.jpg

[**Brad Schulz**](https://www.blogger.com/profile/01852762873611487368)[December 15, 2010 at 11:47 AM](http://bradsruminations.blogspot.com/2010/11/second-in-life-of-query-operator.html?showComment=1292442475646#c6850334943452617578)

@Paul:  
  
By "initialization", do you mean more about the Open() method of the individual iterators?  
  
True, I didn't want to muddy stuff up too much... I wanted to get across the general idea that (as Rob Farley so eloquently states) query plans "suck"... i.e. the iterators request or "pull" data from their children to the right... they don't "push" data from right to left.  
  
A little over a year ago, at least 3 MVP's were puzzling over a query plan at the MSDN forum and couldn't figure out why the counts were the way they were, until the light bulb finally went off over my head as to how they actually worked, and then it all became clear.  
  
I, too, also read plans from right to left to get a feel for the flow of data... but you have to understand the control flow when you get down into more complicated plans like recursive queries or windowed aggregates, for example.  
  
Thanks again, as always for the comment. Your blog post you mentioned promised future posts on the topic... I'm looking forward to it!  
  
--Brad

[Reply](javascript:;)[Delete](https://www.blogger.com/delete-comment.g?blogID=3000721125799986885&postID=6850334943452617578)

1. http://img1.blogblog.com/img/blank.gif

**Anonymous**[December 15, 2010 at 12:30 PM](http://bradsruminations.blogspot.com/2010/11/second-in-life-of-query-operator.html?showComment=1292445059349#c884192260791179244)

Brad,   
  
I have been trying to decide if this post is more misleading than it is pointless. I can't really decide. On one hand you clearly have some things messed up in your head about query plans (thanks for passing those on to the SQL Server community Mr. MVP!). On the other hand though, we are only talking about ways to abstract away what SQL Server is doing for the purpose of better being able to solve problems.   
  
Anyway, your profound and deep paradigm of reading left to right is just awful. I would save this explanation only for the most junior of people that I couldn't get through to any other way. Regardless, it begs the question, if they can't handle the more correct and difficult way of reading query plans then why are you explaining it in the first place?  
  
Sorry for the scathing post but someone had to write it.  
  
Thanks, Jason

[Reply](javascript:;)[Delete](https://www.blogger.com/delete-comment.g?blogID=3000721125799986885&postID=884192260791179244)

1. http://img1.blogblog.com/img/blank.gif

[**Rob Farley**](http://sqlblog.com/blogs/rob_farley)[December 15, 2010 at 2:27 PM](http://bradsruminations.blogspot.com/2010/11/second-in-life-of-query-operator.html?showComment=1292452038876#c1680265608666222456)

Jason,  
  
Many people don't need to understand execution plans at all. For them, getting correct results, and being able to create useful-enough indexes to help performance may get them as far as they need to go.  
  
When people start to read execution plans, they see many resources explain that the data moves from right-to-left, and get stuck there.   
  
Unfortunately if your understanding stops at this point, there are many aspects of plans that remain completely unexplained, particularly around the way that operators know what to do based on operators to their left.  
  
Different people realise the need to consider plans from both directions at different times. I know many people who have found this post from Brad incredibly helpful and enlightening.  
  
Rob

[Reply](javascript:;)[Delete](https://www.blogger.com/delete-comment.g?blogID=3000721125799986885&postID=1680265608666222456)

1. http://img1.blogblog.com/img/blank.gif

**Anonymous**[December 16, 2010 at 11:17 AM](http://bradsruminations.blogspot.com/2010/11/second-in-life-of-query-operator.html?showComment=1292527033412#c1409681682850037834)

Rob,  
  
Off the top of my head I can only think of two operators that affect operators to their right: TOP and filters where the Startup Expression Predicate is specified. There may be others that I forgot or am unaware of, but the point is they are few and far between.   
  
To your point that "Different people realise the need to consider plans from both directions at different times", I agree despite it being very infrequently useful. However, the third line of Brad's post emphatically states that reading from right to left is wrong.  
  
Personally, I think an MVP should be doing things to help the SQL Server community and this post is only going to steer aspiring SQL Server query plan readers down the wrong path.   
  
That's just my two cents though.  
  
Jason

[Reply](javascript:;)[Delete](https://www.blogger.com/delete-comment.g?blogID=3000721125799986885&postID=1409681682850037834)

1. http://img1.blogblog.com/img/blank.gif

[**Rob Farley**](http://sqlblog.com/blogs/rob_farley)[December 16, 2010 at 7:58 PM](http://bradsruminations.blogspot.com/2010/11/second-in-life-of-query-operator.html?showComment=1292558316693#c8219603396487799334)

Jason,  
  
He didn't say it was wrong, he said we said we \*shouldn't\* read plans from right-to-left. He even says that he reads plans from right-to-left to get a feel for the flow of data.  
  
I don't agree with you that it's the wrong path. In fact, Brad is providing an explanation here which is far more complete than many other explanations of how execution plans work.  
  
I'm pleased you've said it's 'just your two cents', but I also think you need to be wary of criticising such an excellent post anonymously, and in such a personal way.  
  
If you'd like to email me on this directly, you can find my email address in the right-hand panel of my own blog. You'll find my name (above) is a link there.  
  
Rob

[Reply](javascript:;)[Delete](https://www.blogger.com/delete-comment.g?blogID=3000721125799986885&postID=8219603396487799334)

1. http://1.bp.blogspot.com/_vPa0QtSASSk/SgJcKyrx9-I/AAAAAAAAAAk/mlghz9uMwAI/S45-s35/biggerpicture.jpg

[**Brad Schulz**](https://www.blogger.com/profile/01852762873611487368)[December 17, 2010 at 12:04 PM](http://bradsruminations.blogspot.com/2010/11/second-in-life-of-query-operator.html?showComment=1292616289496#c7828874856884813009)

Wow, the comments on this post are starting to get longer than the original post itself!  
  
Jason, I wasn't sure at first how to respond to your original post. I appreciate the criticism, but I didn't appreciate the condescending tone... "Mr. MVP"? Gimme a break.  
  
Rob and Paul's comments have already communicated what I want to say on the subject.  
  
Sure, I suppose I could have worded the third paragraph differently... not make it so abrupt... but I wanted to get readers' attention.   
  
As Paul and I had mentioned in these comments, we do generally look at a plan from right to left to get a feel for where the data is going and in what order, but when you get down into detail of record counts processed by the operators and get into plans that involve spools and such, the whole "right-to-left" paradigm only takes you so far.  
  
I wanted to explain, in a hopefully entertaining way, how a query actually works under the hood. I've explained this at SQL user groups, and I can see the revelations in people's eyes as I talk about it, and I've had people express thanks for explaining it.  
  
The first several pages of Craig Freedman's 100-page chapter in Kalen Delaney's "Query Tuning and Optimization" book goes into great detail about how the operators work in the way I described, just as Paul had in his blog.   
  
In short, this is a concept worth knowing, and that's why I wrote about it. We can agree to disagree about its importance, but I'm certainly not doing anything to destroy the SQL Server community... It's not like I'm advocating murder or anything here... I'm surprised and sorry that you felt a need to respond so vehemently.  
  
--Brad

[Reply](javascript:;)[Delete](https://www.blogger.com/delete-comment.g?blogID=3000721125799986885&postID=7828874856884813009)

1. http://3.bp.blogspot.com/_gX1hOs7w6cI/Sax-VVyjB7I/AAAAAAAAAFE/l_bq6y8_rRE/S45-s35/funny-pictures-cat-is-pondering.jpg

[**SDC**](https://www.blogger.com/profile/06074492906736058141)[December 20, 2010 at 12:18 PM](http://bradsruminations.blogspot.com/2010/11/second-in-life-of-query-operator.html?showComment=1292876338416#c1587914728745730654)

I found the article entertaining but also enjoyed the bit of insight - I'll read the query plans in both directions.  
  
After all, there is that expression for somebody who really knows their stuff: 'He/she knows it backwards and forwards'.  
  
Peace and ignore the hatas,  
Steve

[Reply](javascript:;)[Delete](https://www.blogger.com/delete-comment.g?blogID=3000721125799986885&postID=1587914728745730654)

1. http://img1.blogblog.com/img/blank.gif

**Anonymous**[October 31, 2011 at 1:02 PM](http://bradsruminations.blogspot.com/2010/11/second-in-life-of-query-operator.html?showComment=1320091379231#c542915052142231568)

Hello,   
  
Thank you for an excellent read! Just what I needed as I am in the process of trying to fully understand query plans.  
  
I do have a few minor questions, I hope it is not too late to ask.  
  
Quote "NORA: Wake up, Stanley, dear! I need a row for SalesOrderID 47660."  
  
- This implies that the GetNext() method called on the Seek (from Nora/Sort) is called with SalesOrderID as an argument. Is this correctly understood?  
  
- This also implies that the Seek has to wait and can not start about its job before at least one row is returned from the Scan, that is, the Seek is not independent of the scan.  
  
This last point is very confusing as these things are always explained differently in every place I have read, and I would love a   
clarification (the recommendation in the comments to get Delaneys book is noted).  
  
  
Best Regards,  
Geir

[Reply](javascript:;)[Delete](https://www.blogger.com/delete-comment.g?blogID=3000721125799986885&postID=542915052142231568)

1. http://lh3.googleusercontent.com/zFdxGE77vvD2w5xHy6jkVuElKv-U9_9qLkRYK8OnbDeJPtjSZ82UPq5w6hJ-SA=s35

[**LongHairedWeirdo**](https://www.blogger.com/profile/00719702522755422229)[October 2, 2015 at 2:40 PM](http://bradsruminations.blogspot.com/2010/11/second-in-life-of-query-operator.html?showComment=1443822015044#c2640005188617064655)

Thank you for this wonderful post - I've always had an interest in query plans and yes, like many, I assumed that they started in the back and threw the data forward. This is good information, and a whole new model to consider.

[Reply](javascript:;)[Delete](https://www.blogger.com/delete-comment.g?blogID=3000721125799986885&postID=2640005188617064655)

[Replies](javascript:;)

* 1. http://1.bp.blogspot.com/_vPa0QtSASSk/SgJcKyrx9-I/AAAAAAAAAAk/mlghz9uMwAI/S45-s35/biggerpicture.jpg

[**Brad Schulz**](https://www.blogger.com/profile/01852762873611487368)[October 2, 2015 at 4:30 PM](http://bradsruminations.blogspot.com/2010/11/second-in-life-of-query-operator.html?showComment=1443828656239#c6125529886957843969)

Thanks for your reply... I'm glad you found it helpful!

[Delete](https://www.blogger.com/delete-comment.g?blogID=3000721125799986885&postID=6125529886957843969)

[**Reply**](javascript:;)

[**Add comment**](javascript:;)