

SQLIntersection POSTCON01

Make SQL Server Apps Go Faster

Brent Ozar, Jeremiah Peschka, and Kendra Little
Help@BrentOzar.com



BRENT OZAR DOUG LANE JES BORLAND KENDRA LITTLE JEREMIAH PESCHKA

Microsoft CERTIFIED Master SQL Servers 2008 **Microsoft CERTIFIED Professional** **Microsoft CERTIFIED Professional** **Microsoft CERTIFIED Master** SQL Servers 2008 **Microsoft CERTIFIED IT Professional** Database Developer 2008 Database Administrator 2008

MVP Microsoft Most Valuable Professional x 4 **Microsoft Partner** Silver Data Platform **amazon web services** Partner Network

SQL intersection 2 © SQLIntersection. All rights reserved. <http://www.SQLIntersection.com>

SQLintersection

Orlando, Florida – April 2014

www.SQLintersection.com

Make SQL Server apps go faster

- How to Think Like the Engine (Brent)
- Wait Stats: Focusing Your Tuning (Brent)
- “What’s Happening in My SQL Server?” (Jeremiah)
- How to Identify Indexing Problems (Kendra)
- When to Use Filtered Indexes and Indexed Views (Kendra)
- Developer’s Guide to Dangerous Queries (Jeremiah)
- Your Action Plan and How to Get Help



3

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

FAQ, LOL

- We’re at level 200 today
- But ask your questions no matter what!
- Look for your “Montel”
- These chairs get old!
- Get up and walk around the back/sides of the room
- We will take frequent breaks (not just the scheduled ones)
- Watch the timer to know when to get back in your seat

We love questions.

- Asking questions can win you a prize.



4

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>



SQLintersection

Orlando, Florida – April 2014

www.SQLintersection.com

Your top-secret page for the scripts
you'll see today (and more):

<http://BrentOzar.com/go/zoom>

Password: **notslower**



5

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Please evaluate



Don't forget to complete an online evaluation on EventBoard!

Session: POSTCON01

Make SQL Server Apps Go Faster

Your evaluation helps organizers build better conferences,
and helps speakers improve their sessions.



Thank you!



SQLintersection

Orlando, Florida – April 2014

www.SQLintersection.com

SQLintersection Session POSTCON01

How to Think Like the Engine

Brent Ozar

Help@BrentOzar.com



<http://BrentOzar.com/go/engine>

[-] [Table Icon]	dbo.Users
[-] [Folder Icon]	Columns
[Key Icon]	Id (PK, int, not null)
[Table Icon]	Reputation (int, null)
[Table Icon]	CreationDate (datetime, null)
[Table Icon]	DisplayName (nvarchar(40), null)
[Table Icon]	LastAccessDate (datetime, null)
[Table Icon]	WebsiteUrl (nvarchar(200), null)
[Table Icon]	Location (nvarchar(100), null)
[Table Icon]	Age (int, null)
[Table Icon]	AboutMe (nvarchar(max), null)



8

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>



SQLIntersection

Orlando, Florida – April 2014
www.SQLIntersection.com

<http://BrentOzar.com/go/engine>

dbo.Users - Clustered Index

Id	Rep	CreationDate	DisplayName	LastAccessDate	Location	Age	AboutMe
1	2406	7/12/09 10:51 PM	Jeff Atwood	4/1/10 10:35 AM	El Cerrito, CA	39	<a href="http://forums
556	101	7/15/09 9:45 AM	Russ Cam	11/8/09 6:10 PM	UK	NULL	Interested in C#, ASP.NET, s
557	1	7/15/09 9:45 AM	Beto	2/13/10 11:01 AM	NULL	NULL	NULL
558	953	7/15/09 9:46 AM	marco.ragogna	4/1/10 7:41 AM	IT	29	<h2>Software Developer</
559	593	7/15/09 9:46 AM	moobaa	2/18/10 7:14 AM	AU	39	Hey, at least I know 6502!
560	83	7/15/09 9:46 AM	rjstelling	3/4/10 6:47 PM	United Kingdom	30	<h3>Tech geek and hopeles
561	716	7/15/09 9:47 AM	balpha	4/1/10 4:00 PM	NULL	NULL	NULL
562	43	7/15/09 9:47 AM	Mike McClelland	3/16/10 8:45 AM	London	NULL	NULL
563	101	7/15/09 9:48 AM	arturh	10/29/09 2:11 PM	Bremen, Germany	24	NULL
564	960	7/15/09 9:49 AM	Sliff	4/1/10 6:59 AM	UK	30	<p>I'm a Software Enginee
565	121	7/15/09 9:49 AM	Franci Penov	3/5/10 12:13 AM	Kirkland, WA	37	<p>Father, SDE@Microsoft.



From <http://StackOverflow.com>'s Creative Commons Data Dump

rights reserved.
Intersection.com



dbo.Users - Clustered Index

Id	Rep	CreationDate	DisplayName	LastAccessDate	Location	Age	AboutMe
1	2406	7/12/09 10:51 PM	Jeff Atwood	4/1/10 10:35 AM	El Cerrito, CA	39	<a href="http://forums
556	101	7/15/09 9:45 AM	Russ Cam	11/8/09 6:10 PM	UK	NULL	Interested in C#, ASP.NET, s
557	1	7/15/09 9:45 AM	Beto	2/13/10 11:01 AM	NULL	NULL	NULL
558	953	7/15/09 9:46 AM	marco.ragogna	4/1/10 7:41 AM	IT	29	<h2>Software Developer</
559	593	7/15/09 9:46 AM	moobaa	2/18/10 7:14 AM	AU	39	Hey, at least I know 6502!
560	83	7/15/09 9:46 AM	rjstelling	3/4/10 6:47 PM	United Kingdom	30	<h3>Tech geek and hopes
561	716	7/15/09 9:47 AM	balpha	4/1/10 4:00 PM	NULL	NULL	NULL
562	43	7/15/09 9:47 AM	Mike McClelland	3/16/10 8:45 AM	London	NULL	NULL
563	101	7/15/09 9:48 AM	arturh	10/29/09 2:11 PM	Bremen, Germany	24	NULL
564	960	7/15/09 9:49 AM	Sliff	4/1/10 6:59 AM	UK	30	<p>I'm a Software Engineer
565	121	7/15/09 9:49 AM	Franci Penov	3/5/10 12:13 AM	Kirkland, WA	37	<p>Father, SDE@Microsoft

dbo.Users - Clustered Index

Id	Rep	CreationDate	DisplayName	LastAccessDate	Location	Age	AboutMe
566	131	7/15/09 9:50 AM	lil_cain	9/22/09 9:53 PM	Ireland	22	NULL
567	152	7/15/09 9:50 AM	onnodb	4/1/10 6:49 PM	NL	NULL	NULL
568	423	7/15/09 9:50 AM	Pacifika	2/12/10 10:39 AM	Uk	NULL	Web Designer at a communi
569	1	7/15/09 9:51 AM	Michael Jacobsen	2/23/10 2:26 PM	Denmark	34	NULL
570	123	7/15/09 9:51 AM	peSHir	10/27/09 6:56 AM	Vleuten, NL	36	Senior .NET software develo
571	139	7/15/09 9:52 AM	Tundey	10/8/09 6:27 PM	US	32	Software engineer by day. S
572	113	7/15/09 9:52 AM	Kaius	3/16/10 3:51 PM	Kilkenny, Ireland	NULL	I am a .NET software engine
573	11	7/15/09 9:52 AM	Damian Powell	3/22/10 11:41 AM	Redditch, UK	34	Programmer since the age c
574	101	7/15/09 9:53 AM		3/20/10 1:07 PM	NULL	NULL	NULL
575	23	7/15/09 9:53 AM	rmontagud	3/30/10 10:05 PM	Valencia, Spain	28	Just a random person
576	11	7/15/09 9:53 AM	L.R.	3/23/10 6:20 PM	Czech Republic	28	NULL
577	329	7/15/09 9:54 AM	Jonas	2/28/10 4:37 PM	SE	24	A happy programmer! :-)
578	123	7/15/09 9:54 AM	IIDan	9/23/09 9:54 AM	Italy	40	Living in Italy, Cocoa progra
579	196	7/15/09 9:54 AM	kartikmohta	3/16/10 6:14 PM	India	23	NULL
580	215	7/15/09 9:54 AM	n00ki3	8/22/09 11:58 AM	Gelsenkirchen	22	I am a student at Bioinform
581	101	7/15/09 9:55 AM	Wayne Koorts	2/18/10 9:40 AM	NZ	23	Software development and
582	1	7/15/09 9:55 AM	kender	2/5/10 10:24 AM	PL	30	Programmer by heart, curre
583	101	7/15/09 9:56 AM	Mauli	12/15/09 7:06 AM	Munich	30	My favourite language is Pyt
584	101	7/15/09 9:56 AM	nickd	3/27/10 10:04 AM	Ireland	44	.NET developer
585	384	7/15/09 9:56 AM	Hemant	3/29/10 8:44 AM	India	28	<a href="http://www.googl
586	1931	7/15/09 9:58 AM	pgs	3/19/10 6:23 AM	Sydney, Australia	43	Just being myself.
587	101	7/15/09 9:58 AM		7/15/09 9:58 AM	NULL	NULL	NULL
588	1	7/15/09 9:58 AM		11/7/09 11:28 AM	US	NULL	NULL
589	31	7/15/09 9:58 AM	Nikolaas	12/22/09 8:07 AM	NULL	35	NULL
590	526	7/15/09 9:59 AM	Mark Thalman	3/29/10 7:56 PM	Long Island, NY	43	Smile.
591	3552	7/15/09 9:59 AM	innaM	3/28/10 9:56 AM	NULL	NULL	NULL
592	155	7/15/09 10:00 AM	Richard Morgan	3/29/10 11:59 AM	Columbus, Ohio	38	C# Developer
593	258	7/15/09 10:00 AM	rq	4/1/10 8:48 PM	Madrid, Spain	32	I code mostly in Python and
594	1	7/15/09 10:00 AM	Louis Haußknecht	7/15/09 10:00 AM	NULL	NULL	NULL
595	1	7/15/09 10:01 AM	Salaryman	9/1/09 1:39 AM	NULL	NULL	NULL
596	700	7/15/09 10:01 AM	stukelly	3/31/10 7:47 AM	Preston, UK	30	I work on the help desk for
597	101	7/15/09 10:02 AM	elo80ka	7/15/09 10:02 AM	Port Harcourt, Nigeria	29	NULL
598	101	7/15/09 10:03 AM		2/8/10 4:13 PM	NULL	NULL	NULL
599	111	7/15/09 10:05 AM	SWB	3/31/10 11:41 PM	New Albany, Indiana, U	36	NULL
600	101	7/15/09 10:05 AM	DrFloyd5	10/1/09 4:44 AM	NULL	35	<P>Currently programming

dbo.Users - Clustered Index

Id	Rep	CreationDate	DisplayName	LastAccessDate	Location	Age	AboutMe
601	101	7/15/09 10:07 AM	Phil Jenkins	1/20/10 9:39 AM	Manchester, UK	26	I'm a software consultant fo
602	371	7/15/09 10:08 AM	Will	11/30/09 9:59 PM	NULL	30	NULL
603	1	7/15/09 10:08 AM	Jeras	12/6/09 2:24 PM	United States of America	26	Just a simple programmer, t
604	126	7/15/09 10:09 AM	Ben Williams	4/1/10 10:26 AM	Boston, MA	NULL	NULL
605	323	7/15/09 10:11 AM	Thomas Owens	3/20/10 6:02 PM	Rochester, New York, U	21	I am a software engineering
606	174	7/15/09 10:11 AM	PhilPursglove	4/1/10 3:51 PM	Cambridge, UK	35	Not poems and rubbish - Sc
607	106	7/15/09 10:11 AM	Holgerwa	2/12/10 6:33 PM	Germany	42	NULL
608	121	7/15/09 10:15 AM	Jonas	9/27/09 12:24 PM	Finland	NULL	NULL
609	178	7/15/09 10:15 AM	chrish	12/16/09 7:26 PM	US	38	I am a C++ developer for Ir
610	71	7/15/09 10:16 AM	Charles	3/9/10 8:28 PM	Washington, D.C.	29	Wow, it's on the Internet? It
611	103	7/15/09 10:17 AM	rampion	9/15/09 1:17 PM	Silver Spring, MD	30	Mathematician and research
612	101	7/15/09 10:17 AM	Rob Burke	3/22/10 3:24 PM	Dublin, Ireland	22	Navision / C# developer wit
613	172	7/15/09 10:17 AM	Colen	2/2/10 1:40 AM	San Jose, CA	29	Win32 software developer.
614	116	7/15/09 10:18 AM	cadrian	12/16/09 1:16 PM	Bart (France)	35	My website says it all. Go th
615	1	7/15/09 10:19 AM	Melle	7/26/09 8:48 AM	Amsterdam	NULL	NULL
616	198	7/15/09 10:20 AM	McDowell	10/3/09 9:45 PM	\$HOME	36	Still looking for the ANY key
617	281	7/15/09 10:20 AM	David Basarab	3/23/10 1:07 PM	Medina, OH	31	David Basarab</hr
618	1	7/15/09 10:21 AM	Marlen	7/15/09 10:21 AM	NL	30	I am a software programme
619	138	7/15/09 10:21 AM	Nick Berardi	3/6/10 12:43 PM	Phoenixville, PA	30	Nicholas Berardi has a BS in
620	636	7/15/09 10:22 AM	Damien	3/18/10 2:13 PM	London, UK	23	Interested in:
621	101	7/15/09 10:23 AM	empi	11/10/09 2:47 PM	NULL	NULL	NULL
622	376	7/15/09 10:24 AM	Spedge	9/30/09 12:05 PM	NULL	NULL	NULL
623	101	7/15/09 10:26 AM	Fionn	7/15/09 10:26 AM	NULL	NULL	NULL
624	111	7/15/09 10:26 AM	Ramesh Soni	3/30/10 6:31 AM	India	27	NULL
625	101	7/15/09 10:27 AM	Alo	3/27/10 10:19 AM	NULL	NULL	NULL
626	206	7/15/09 10:27 AM	Prashant	2/22/10 5:47 PM	Delhi, India	23	A web application developer
627	198	7/15/09 10:27 AM		4/1/10 4:06 PM	NULL	NULL	NULL
628	1	7/15/09 10:28 AM	tjlevine	8/17/09 10:11 AM	Bay Area, CA	18	I'm a kid from the Bay Area
629	101	7/15/09 10:28 AM	Zoomzoom83	7/15/09 10:28 AM	NULL	26	NULL
630	853	7/15/09 10:29 AM	Vinko Vrsalovic	2/20/10 10:41 PM	The sky	67	A generalist. Or, better put,
631	101	7/15/09 10:29 AM	palm3D	1/29/10 11:14 AM	BE	27	Computer scientist
632	206	7/15/09 10:29 AM	SilentGhost	3/29/10 10:14 AM	NULL	NULL	NULL
633	166	7/15/09 10:29 AM	Vincent Buck	10/2/09 4:46 PM	Brussels, Belgium	43	NULL
634	101	7/15/09 10:31 AM	Paul Brown	3/25/10 2:07 PM	NULL	NULL	NULL
635	572	7/15/09 10:31 AM		4/1/10 2:29 PM	NULL	NULL	NULL

dbo.Users - Clustered Index

Id	Rep	CreationDate	DisplayName	LastAccessDate	Location	Age	AboutMe
636	1	7/15/09 10:31 AM	Tobias Gies	2/13/10 8:37 PM	Cologne, Germany	19	NULL
637	1	7/15/09 10:32 AM	Kitt Basch	4/1/10 9:12 AM	NULL	NULL	NULL
638	101	7/15/09 10:32 AM		7/15/09 10:32 AM	NULL	NULL	NULL
639	106	7/15/09 10:32 AM	Dave Van den Ey	2/15/10 7:29 PM	Belgium	34	NULL
640	101	7/15/09 10:33 AM	animajo	3/18/10 8:27 AM	Salzburg, Austria	32	yes
641	11	7/15/09 10:33 AM	StuperUser	3/18/10 1:47 PM	London	24	A junior .NET and SQL deve
642	101	7/15/09 10:36 AM	aherrick	7/15/09 10:36 AM	Indianapolis	23	I like to code.
643	101	7/15/09 10:36 AM	Sarcastic	8/18/09 1:14 PM	Edinburgh, UK	30	I'm a miserable git.
644	122	7/15/09 10:37 AM	spaetzel	9/9/09 2:16 PM	CA	27	Web developer, designer and
645	295	7/15/09 10:37 AM	Matthew Schinck	3/30/10 10:41 AM	Adelaide, Australia	34	Programmer.
646	196	7/15/09 10:38 AM	jao	3/26/10 4:59 PM	NULL	29	NULL
647	329	7/15/09 10:39 AM	Pawka	3/23/10 6:11 PM	Lithuania // Kaunas	24	Student. Web developer. C
648	106	7/15/09 10:40 AM	Jon Cram	12/11/09 11:12 PM	GB	32	<p> I'm <a href="http://w
649	1	7/15/09 10:40 AM		11/28/09 2:22 PM	NULL	NULL	NULL
650	731	7/15/09 10:41 AM	Ziggamorph	3/22/10 10:57 AM	Cambridge, UK	NULL	NULL
651	1	7/15/09 10:41 AM	Worthless Genius	2/23/10 9:23 PM	Mount Savage, MD, US	25	I am currently a student at ,
652	323	7/15/09 10:41 AM	X-Istence	4/1/10 9:48 PM	::1	21	Bert JW Regeer is a softwar
653	531	7/15/09 10:42 AM	person-b	3/31/10 8:27 PM	Stirling, Scotland	13	I'm a hobbyist C#, Python a
654	111	7/15/09 10:43 AM	mwore	3/26/10 3:40 PM	Slovenia	24	Freelance software develop
655	150	7/15/09 10:43 AM	dev	3/24/10 9:35 AM	Mumbai	24	Just a guy who is hacking st
656	101	7/15/09 10:44 AM		7/28/09 10:53 PM	NULL	NULL	NULL
657	413	7/15/09 10:44 AM	alex	4/1/10 1:46 PM	sunshine coast, queens	24	I am a web developer. I sad
658	111	7/15/09 10:45 AM	Anders	12/4/09 8:33 AM	Norway	26	NULL
659	347	7/15/09 10:45 AM	Stephen	3/11/10 4:17 PM	Ireland	21	Student coder
660	1256	7/15/09 10:46 AM	Richard	3/26/10 9:08 AM	GB	41	<p>A professional develop
661	101	7/15/09 10:46 AM		7/15/09 10:46 AM	NULL	NULL	NULL
662	101	7/15/09 10:46 AM	KnackeredCoder	9/30/09 12:09 PM	NULL	NULL	NULL
663	111	7/15/09 10:46 AM	Rob G	1/25/10 8:40 PM	GB	32	I can has coding?
664	475	7/15/09 10:47 AM	Piotr Dobrogost	11/11/09 8:49 PM	Poland	32	C++ is the king
665	280	7/15/09 10:47 AM	Tomas Sedovic	3/18/10 11:28 PM	Brno, Czech Republic	24	NULL
666	101	7/15/09 10:48 AM	Matthew Savage	10/26/09 2:27 PM	Canberra, Australia	24	NULL
667	223	7/15/09 10:48 AM	hasseg	3/21/10 6:33 PM	FI	NULL	<p> My name is Ali Rantak
668	101	7/15/09 10:51 AM	mtruesdell	8/28/09 7:51 PM	US	34	Java Servlet programmer lo
669	101	7/15/09 10:51 AM	Tim Saunders	9/6/09 9:24 PM	London	34	<p>Development Director (
670	306	7/15/09 10:51 AM	David M	8/1/09 7:29 PM	GB	42	Software developer and bas

dbo.Users - IX_LastAccessDate

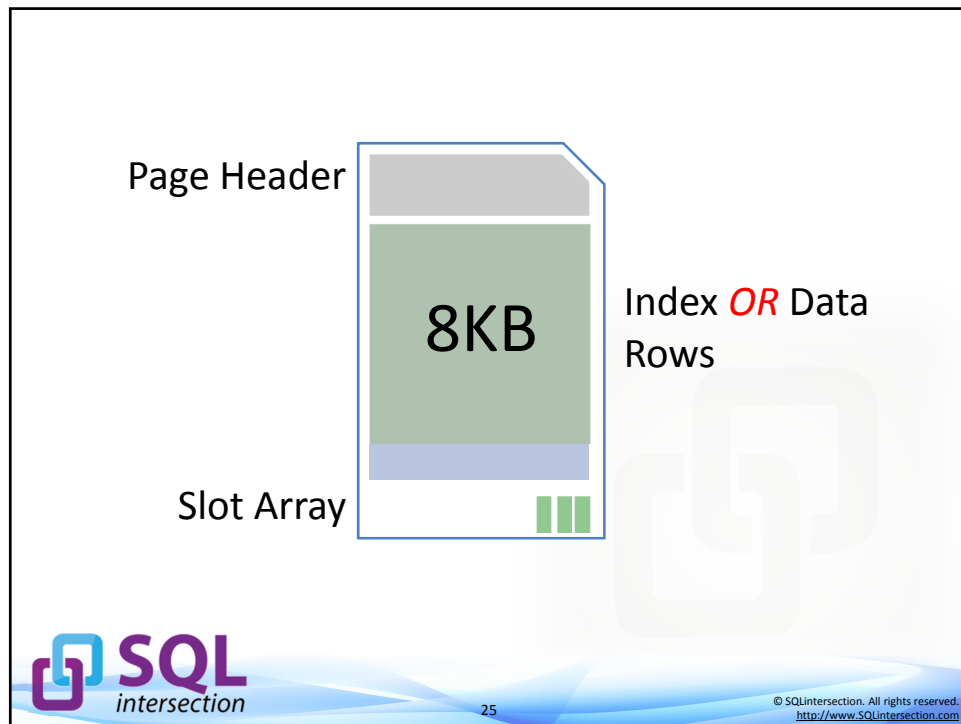
LastAccessDate	Id	LastAccessDate	Id	LastAccessDate	Id	LastAccessDate	Id
7/31/08 12:00 AM	-1	7/15/09 8:53 AM	445	7/15/09 9:10 PM	200	8/11/09 7:17 PM	39
7/15/09 7:08 AM	22	7/15/09 8:58 AM	457	7/16/09 6:22 AM	678	8/12/09 2:54 PM	943
7/15/09 7:10 AM	33	7/15/09 9:17 AM	501	7/17/09 2:30 AM	131	8/13/09 4:26 PM	364
7/15/09 7:11 AM	40	7/15/09 9:28 AM	524	7/17/09 9:30 AM	297	8/15/09 5:03 PM	910
7/15/09 7:11 AM	41	7/15/09 9:30 AM	527	7/17/09 8:43 PM	998	8/17/09 8:42 AM	202
7/15/09 7:11 AM	44	7/15/09 9:58 AM	587	7/18/09 12:38 PM	394	8/17/09 10:11 AM	628
7/15/09 7:12 AM	52	7/15/09 10:00 AM	594	7/18/09 2:15 PM	924	8/17/09 10:33 AM	157
7/15/09 7:13 AM	64	7/15/09 10:02 AM	597	7/19/09 10:26 PM	336	8/17/09 4:24 PM	1006
7/15/09 7:13 AM	65	7/15/09 10:21 AM	618	7/20/09 1:06 PM	849	8/18/09 8:06 AM	511
7/15/09 7:14 AM	68	7/15/09 10:25 AM	347	7/21/09 7:22 AM	881	8/18/09 9:00 AM	262
7/15/09 7:15 AM	73	7/15/09 10:26 AM	623	7/23/09 11:53 AM	503	8/18/09 9:43 AM	210
7/15/09 7:17 AM	87	7/15/09 10:28 AM	629	7/23/09 12:56 PM	446	8/18/09 10:22 AM	673
7/15/09 7:18 AM	92	7/15/09 10:32 AM	638	7/24/09 12:15 AM	407	8/18/09 1:05 PM	959
7/15/09 7:20 AM	110	7/15/09 10:36 AM	642	7/24/09 12:08 PM	488	8/18/09 1:14 PM	643
7/15/09 7:26 AM	139	7/15/09 10:46 AM	661	7/24/09 3:17 PM	144	8/18/09 1:32 PM	525
7/15/09 7:36 AM	173	7/15/09 10:59 AM	686	7/26/09 8:48 AM	615	8/18/09 1:38 PM	159
7/15/09 7:38 AM	179	7/15/09 11:14 AM	722	7/27/09 2:27 PM	337	8/18/09 2:28 PM	690
7/15/09 7:46 AM	223	7/15/09 11:26 AM	756	7/28/09 3:17 AM	174	8/19/09 9:22 AM	750
7/15/09 7:57 AM	258	7/15/09 11:48 AM	803	7/28/09 12:34 PM	118	8/19/09 2:46 PM	887
7/15/09 7:58 AM	259	7/15/09 11:50 AM	681	7/28/09 9:28 PM	201	8/20/09 3:45 AM	749
7/15/09 8:03 AM	279	7/15/09 11:51 AM	808	7/28/09 10:53 PM	656	8/20/09 8:29 AM	268
7/15/09 8:06 AM	286	7/15/09 12:00 PM	822	7/29/09 12:06 PM	751	8/20/09 10:32 AM	901
7/15/09 8:12 AM	305	7/15/09 12:07 PM	841	7/30/09 12:26 PM	754	8/20/09 12:05 PM	752
7/15/09 8:13 AM	313	7/15/09 12:07 PM	842	7/31/09 6:36 PM	11	8/20/09 3:03 PM	272
7/15/09 8:14 AM	320	7/15/09 12:18 PM	878	8/1/09 12:09 PM	775	8/21/09 7:29 PM	487
7/15/09 8:16 AM	323	7/15/09 12:18 PM	779	8/1/09 7:29 PM	670	8/22/09 11:58 AM	580
7/15/09 8:25 AM	349	7/15/09 12:47 PM	973	8/1/09 11:33 PM	85	8/24/09 11:22 AM	56
7/15/09 8:26 AM	354	7/15/09 12:48 PM	819	8/3/09 10:13 PM	300	8/24/09 11:33 PM	889
7/15/09 8:27 AM	97	7/15/09 12:52 PM	987	8/5/09 12:06 PM	918	8/25/09 8:57 PM	150
7/15/09 8:27 AM	363	7/15/09 12:54 PM	992	8/6/09 5:46 AM	726	8/26/09 2:43 AM	851
7/15/09 8:33 AM	384	7/15/09 1:28 PM	931	8/6/09 10:42 AM	370	8/26/09 2:53 AM	117
7/15/09 8:37 AM	402	7/15/09 2:19 PM	728	8/6/09 12:37 PM	398	8/26/09 9:59 AM	546
7/15/09 8:47 AM	431	7/15/09 2:20 PM	217	8/9/09 5:28 AM	45	8/27/09 3:54 AM	239
7/15/09 8:52 AM	440	7/15/09 4:23 PM	975	8/10/09 12:25 AM	389	8/27/09 2:10 PM	271

dbo.Users - IX_LastAccessDate_ID_DisplayName_Age

LastAccessDate	Id	DisplayName	Age	LastAccessDate	Id	DisplayName	Age
7/31/08 12:00 AM	-1	Community	1	8/18/09 10:22 AM	673		NULL
7/15/09 9:58 AM	587		NULL	8/18/09 1:14 PM	643	Sarcastic	30
7/15/09 10:00 AM	594	Louis Haußknecht	NULL	8/18/09 2:28 PM	690	msh	31
7/15/09 10:02 AM	597	elo80ka	29	8/19/09 9:22 AM	750	Cyphus	30
7/15/09 10:21 AM	618	Marien	30	8/20/09 3:45 AM	749		NULL
7/15/09 10:26 AM	623	Fionn	NULL	8/20/09 12:05 PM	752	Sean Taylor	27
7/15/09 10:28 AM	629	Zoomzoom83	26	8/22/09 11:58 AM	580	n00ki3	22
7/15/09 10:32 AM	638		NULL	8/26/09 2:43 AM	851	Tim Howland	41
7/15/09 10:36 AM	642	aherrick	23	8/26/09 9:59 AM	546	hitec	27
7/15/09 10:46 AM	661		NULL	8/28/09 7:51 PM	668	mtruesdell	34
7/15/09 10:59 AM	686	urini	38	9/1/09 1:39 AM	595	Salaryman	NULL
7/15/09 11:14 AM	722		NULL	9/6/09 9:24 PM	669	Tim Saunders	34
7/15/09 11:26 AM	756	suhair	29	9/8/09 2:56 AM	869	Malke	NULL
7/15/09 11:48 AM	803	Mike H	NULL	9/9/09 2:16 PM	644	spaetzel	27
7/15/09 11:50 AM	681	Foom	NULL	9/14/09 2:19 PM	707	17 of 26	38
7/15/09 11:51 AM	808		NULL	9/15/09 1:17 PM	611	rampion	30
7/15/09 12:00 PM	822	atoumey	26	9/18/09 10:56 AM	848	baxter	26
7/15/09 12:07 PM	841	Nick Bradbury	NULL	9/22/09 9:53 PM	566	lil_cain	22
7/15/09 12:07 PM	842	Ian Oxley	33	9/23/09 9:54 AM	578	IIDan	40
7/15/09 12:18 PM	878		NULL	9/27/09 12:24 PM	608	Jonas	NULL
7/15/09 12:18 PM	779	Don Kitchen	32	9/30/09 8:53 AM	852	Manne	30
7/15/09 12:48 PM	819	Sebastian Dietz	31	9/30/09 12:05 PM	622	Spedge	NULL
7/15/09 2:19 PM	728	Matt Price	29	9/30/09 12:09 PM	662	KnackeredCoder	NULL
7/16/09 6:22 AM	678		NULL	10/1/09 4:44 AM	600	DrFloyd5	35
7/20/09 1:06 PM	849	valters	31	10/2/09 4:46 PM	633	Vincent Buck	43
7/21/09 7:22 AM	881	wai	NULL	10/3/09 9:45 PM	616	McDowell	36
7/26/09 8:48 AM	615	Melle	NULL	10/8/09 6:27 PM	571	Tundey	32
7/28/09 10:53 PM	656		NULL	10/13/09 5:34 PM	858	jj33	32
7/29/09 12:06 PM	751	Sruly	24	10/15/09 6:11 AM	698	mishac	90
7/30/09 12:26 PM	754	Baishampayan G	27	10/20/09 11:48 AM	691		NULL
8/1/09 12:09 PM	775	Richard Nichols	30	10/21/09 4:28 PM	800	jrom	23
8/1/09 7:29 PM	670	David M	42	10/26/09 2:27 PM	666	Matthew Savage	24
8/6/09 5:46 AM	726	Refracted Paladin	30	10/27/09 6:56 AM	570	peSHir	36
8/17/09 10:11 AM	628	tjlevine	18	10/28/09 7:27 PM	705	TheTXI	25

dbo.Users - IX_LastAccessDate_Id_DisplayName_Age

LastAccessDate	Id	DisplayName	Age	LastAccessDate	Id	DisplayName	Age
10/28/09 7:48 PM	674	ruddzw	22	12/12/09 12:49 AM	708	hernan43	32
10/29/09 2:11 PM	563	arturh	24	12/12/09 2:38 AM	771	Ben Lever	30
10/30/09 2:57 PM	873	Hath	28	12/12/09 5:12 PM	543	DavidWhitney	25
11/1/09 8:06 PM	763	Tom	NULL	12/14/09 3:40 PM	748	Jamie	32
11/3/09 7:55 PM	836		NULL	12/14/09 11:07 PM	865	EndangeredMassa	24
11/4/09 3:29 PM	877	digiduck	23	12/15/09 7:06 AM	583	Mauli	30
11/7/09 11:28 AM	588		NULL	12/15/09 12:28 PM	729	Mike Cornell	35
11/8/09 6:10 PM	556	Russ Cam	NULL	12/15/09 1:58 PM	872		NULL
11/9/09 9:47 PM	846	Luís Oliveira	NULL	12/15/09 3:27 PM	843	Steve Beedie	32
11/9/09 9:54 PM	753	Erv Walter	35	12/16/09 1:16 PM	614	cadrian	35
11/10/09 2:47 PM	621	empi	NULL	12/16/09 7:26 PM	609	chrish	38
11/11/09 8:49 PM	664	Piotr Dobrogost	32	12/20/09 5:14 PM	688	databyss	29
11/12/09 5:13 PM	847	Ian Potter	24	12/21/09 10:33 AM	773	Rodrigo Guerreirc	29
11/24/09 9:17 AM	713	temp	NULL	12/22/09 8:07 AM	589	Nikolaas	35
11/28/09 8:25 AM	734	derigel	31	12/25/09 2:04 PM	867	longeasy	NULL
11/28/09 2:22 PM	649		NULL	12/27/09 8:55 PM	825	James Muscat	22
11/29/09 7:20 AM	552	Exodus	25	12/30/09 11:12 AM	805	Maximilian	NULL
11/30/09 9:45 AM	813	7alwagy	26	12/31/09 5:04 AM	737	Nick	24
11/30/09 9:59 PM	602	Will	30	1/11/10 5:37 AM	835	bjtitus	19
12/1/09 12:54 PM	853	keith	26	1/11/10 7:51 AM	802	jkp	NULL
12/1/09 4:37 PM	540	DanSingerman	NULL	1/13/10 11:19 AM	850	Ionel Bratianu	24
12/2/09 8:33 AM	796	David Maymudes	40	1/16/10 9:46 PM	782	dF	31
12/2/09 3:21 PM	721	Otherside	27	1/17/10 10:45 PM	824	skfd	NULL
12/4/09 5:42 AM	743		NULL	1/19/10 9:15 PM	818	Nicolas Cadilhac	39
12/4/09 8:33 AM	658	Anders	26	1/19/10 10:54 PM	538	cgreeno	NULL
12/6/09 2:24 PM	603	Jeras	26	1/20/10 9:39 AM	601	Phil Jenkins	26
12/6/09 3:00 PM	747	Rick	NULL	1/23/10 1:45 AM	812	sestocker	26
12/8/09 6:11 PM	875	Nick Haddad	32	1/25/10 9:42 AM	870	Fabian Steeg	29
12/10/09 12:36 PM	809		NULL	1/25/10 7:10 PM	537	staffan	36
12/10/09 3:05 PM	814	Kevin	32	1/25/10 8:40 PM	663	Rob G	32
12/11/09 1:45 PM	876	Darren Greaves	NULL	1/26/10 7:48 PM	695	David Negron	35
12/11/09 10:40 PM	838	BenMaddox	25	1/26/10 9:49 PM	706	Dawie Strauss	NULL
12/11/09 10:45 PM	871	JPrescottSanders	32	1/27/10 1:36 AM	671	Sam Jones	NULL
12/11/09 11:12 PM	648	Jon Cram	32	1/27/10 3:39 PM	711	DMMcKinnon	25



DBCC IND lists a table's pages.

```

/*
To find the pages used by a table, use DBCC IND with 3 parameters:
the database name, the table name, and -1 for all indexes:
*/
DBCC IND('StackOverflow', Users, -1);

```

	PageFID	PagePID	IAMFID	IAMPID	ObjectID	IndexID	PartitionNumber	PartitionID	Iam_chain_type	PageType	IndexLevel	NextPageFID	NextPagePID
1	1	1231497	NULL	NULL	494624805	1	1	72057594050707456	In-row data	10	NULL	1	1022487
2	1	511269	1	1231497	494624805	1	1	72057594050707456	In-row data	1	0	0	0
3	1	511270	1	1231497	494624805	1	1	72057594050707456	In-row data	1	0	1	511269
4	1	1022480	1	1231497	494624805	1	1	72057594050707456	In-row data	1	0	1	511270
5	1	1022481	1	1231497	494624805	1	1	72057594050707456	In-row data	1	0	1	1022480
6	1	1022482	1	1231497	494624805	1	1	72057594050707456	In-row data	1	0	1	1022481
7	1	1022483	1	1231497	494624805	1	1	72057594050707456	In-row data	1	0	1	1022482
8	1	1022485	1	1231497	494624805	1	1	72057594050707456	In-row data	1	0	1	1022483
9	1	1022486	1	1231497	494624805	1	1	72057594050707456	In-row data	1	0	1	1022485
10	1	1022487	NULL	NULL	494624805	1	1	72057594050707456	In-row data	10	NULL	0	0
11	1	883696	1	1022487	494624805	1	1	72057594050707456	In-row data	1	0	1	1022486
12	1	883697	1	1022487	494624805	1	1	72057594050707456	In-row data	1	0	1	883696
13	1	883698	1	1022487	494624805	1	1	72057594050707456	In-row data	1	0	1	883697
14	1	883699	1	1022487	494624805	1	1	72057594050707456	In-row data	1	0	1	883698
15	1	883700	1	1022487	494624805	1	1	72057594050707456	In-row data	1	0	1	883699
16	1	883701	1	1022487	494624805	1	1	72057594050707456	In-row data	1	0	1	883700
17	1	883702	1	1022487	494624805	1	1	72057594050707456	In-row data	1	0	1	883701
18	1	883703	1	1022487	494624805	1	1	72057594050707456	In-row data	1	0	1	883702

SQLIntersection

Orlando, Florida – April 2014
www.SQLIntersection.com

```
/*
To see the contents of a page, enable trace flag 3604 for your session,
and then call DBCC PAGE with 4 parameters:
the database name, file name, page number, and 3 to see the details.
*/
DBCC TRACEON(3604);
DBCC PAGE('StackOverflow', 1, 1022481, 3);
DBCC TRACEOFF(3604);

```

100 %

Messages

DBCC execution completed. If DBCC printed error messages, contact your system administrator.

PAGE: (1:1022481)

BUFFER:

BUF @0x000000181167080

bpage = 0x000000016E23A000 bhash = 0x0000000000000000 bpageNo = (1:1022481)
bobjid = 7 bpreferences = 0 bcpucticks = 0
bsampleCount = 0 bUse1 = 543 bstat = 0x9
blog = 0x15a bnext = 0x0000000000000000

PAGE HEADER:

Page @0x00000016E23A000

m_pageId = (1:1022481) m_headerVersion = 1 m_type = 1
m_typeFlagBits = 0x80 m_level = 0 m_flagBits = 0x8200
m_objId (AllocUnitId.idObj) = 302 m_indexId (AllocUnitId.idInd) = 256
Metadata: AllocUnitId = 72057594057719808 Metadata: IndexId = 1
Metadata: PartitionId = 72057594050707456 m_prevPage = (1:1022482) m_nextPage = (1:1022480)
Metadata: ObjectId = 494624805 m_slotCnt = 88 m_freeCnt = 55
m_inlen = 9 m_reservedCnt = 0 m_lsn = (294:22646:41)
m_freeData = 7961 m_xdesId = (0:0) m_ghostRecCnt = 0
m_xactReserved = 0 DB Frag ID = 1
m_tornBits = -1554435873

Allocation Status

GAM (1:1022464) = ALLOCATED SGAM (1:1022465) = NOT ALLOCATED
PFS (1:1019088) = 0x60 MIXED_EXT ALLOCATED _PCT_FULL DIFF (1:1022470) = CHANGED
ML (1:1022471) = NOT MTN LOGGED

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

```
Slot 1 Column 1 Offset 0x4 Length 4 Length (physical) 4
Id = 39257

Slot 1 Column 2 Offset 0x8 Length 4 Length (physical) 4
Reputation = 1

Slot 1 Column 3 Offset 0xc Length 8 Length (physical) 8
CreationDate = 2010-03-30 19:09:35.540

Slot 1 Column 4 Offset 0x3a Length 10 Length (physical) 10
DisplayName = AndyN

Slot 1 Column 5 Offset 0x14 Length 8 Length (physical) 8
LastAccessDate = 2010-04-01 02:20:09.317

Slot 1 Column 6 Offset 0x0 Length 0 Length (physical) 0
WebsiteUrl = [NULL]

Slot 1 Column 7 Offset 0x44 Length 34 Length (physical) 34
Location = Bay Area, CA, USA

Slot 1 Column 8 Offset 0x1c Length 4 Length (physical) 4
Age = 37

AboutMe = [BLOB Inline Data] Slot 1 Column 9 Offset 0x66 Length 84 Length (physical) 84

627612E0: 57006100 74006300 68006900 6e006700 tW.a.t.c.h.i.n.g.
627612F0: 20007000 72006f00 67006500 73007300 t.p.r.o.g.e.s.s.
62761300: 20006200 61007200 73002000 66006f00 t.b.a.r.s..f.o.
62761310: 72002000 6d006f00 6e006500 79002000 tr..m.o.n.e.y..
62761320: 73006900 6e006300 65002000 31003900 ts.i.n.c.e..l.9.
62761330: 39003400 ++++++9.4.
```

reserved.
ion.com



SQLIntersection

Orlando, Florida – April 2014

www.SQLIntersection.com

You: SQL Server.
Me: end user.

Tarzan write query good.



First query:

```
SELECT *  
FROM dbo.Users
```



30

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>



Your Execution Plan

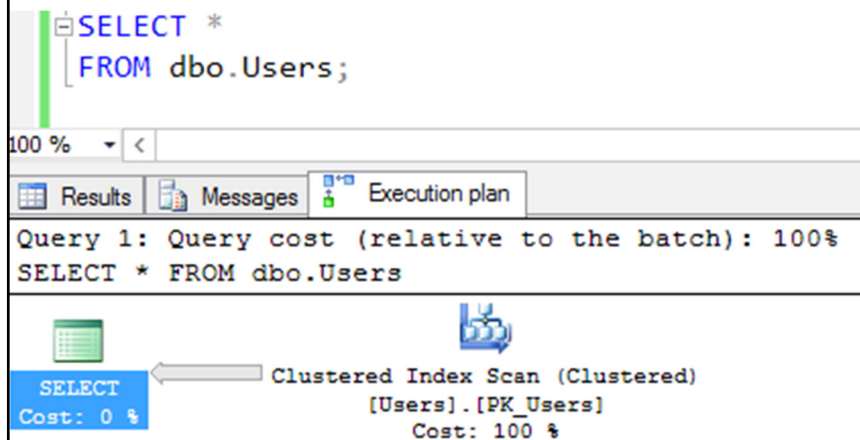
1. Read all the Users records



31

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

SQL Server's Execution Plan



32

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Next up

```
SELECT Id
FROM dbo.Users
WHERE LastAccessDate > '7/1/2010'
```



33

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Your Execution Plan

1. Read all the Users records,
only “saying” the Ids for
the ones with LastAccessDate > '7/1/2010'



34

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

SQLIntersection

Orlando, Florida – April 2014

www.SQLIntersection.com

SQL Server's Execution Plan

The screenshot shows the SQL Server Enterprise Manager interface. The query editor contains the following SQL statement:

```
SELECT Id
FROM dbo.Users
WHERE LastAccessDate > '7/1/2010';
```

The 'Execution plan' tab is selected. The query cost is 100%. The execution plan shows a 'Clustered Index Scan (Clustered)' on the 'PK_Users' index of the 'dbo.Users' table. The cost is 100%.

SQL intersection

35

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

The screenshot compares two execution plans side-by-side. Both queries are shown in the query editor:

```
SELECT *
FROM dbo.Users;
```

```
SELECT Id
FROM dbo.Users
WHERE LastAccessDate > '7/1/2010';
```

Both queries have a cost of 100%. The execution plan for the first query (SELECT *) shows a 'Clustered Index Scan (Clustered)' on the 'PK_Users' index. The 'Estimated Subtree Cost' is 26.0171, which is circled in red. The 'Estimated Number of Rows' is 1.

The execution plan for the second query (SELECT Id WHERE LastAccessDate > '7/1/2010') also shows a 'Clustered Index Scan (Clustered)' on the 'PK_Users' index. The 'Estimated Subtree Cost' is 26.0171, which is circled in red. The 'Estimated Number of Rows' is 1.

SQL intersection

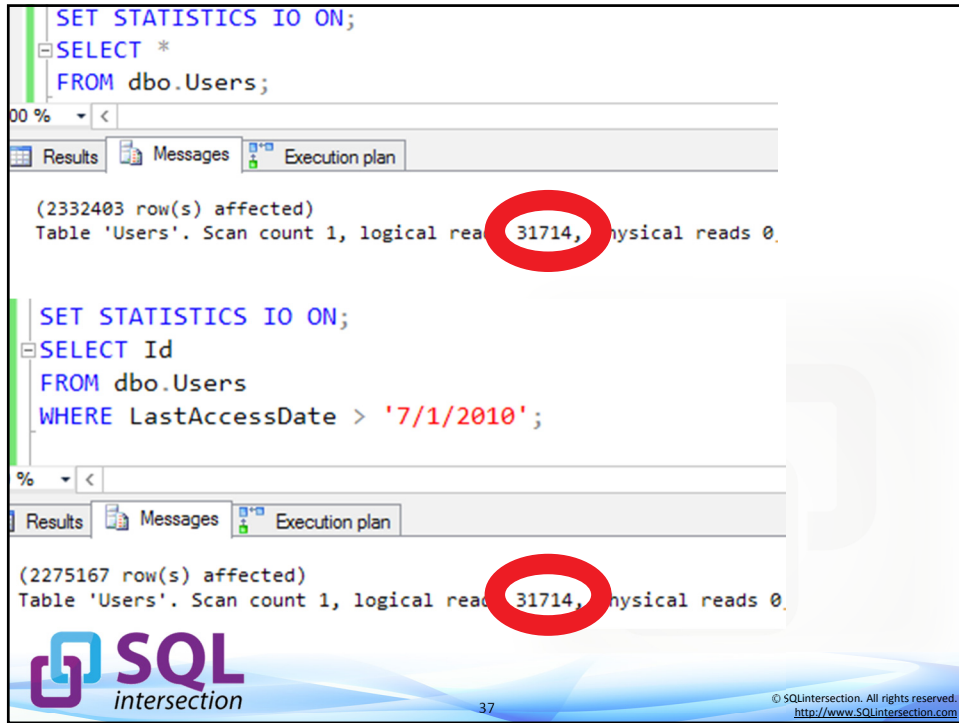
36

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

SQLIntersection

Orlando, Florida – April 2014

www.SQLIntersection.com



```
SET STATISTICS IO ON;  
SELECT *  
FROM dbo.Users;
```

(2332403 row(s) affected)
Table 'Users'. Scan count 1, logical reads 31714, physical reads 0.

```
SET STATISTICS IO ON;  
SELECT Id  
FROM dbo.Users  
WHERE LastAccessDate > '7/1/2010';
```

(2275167 row(s) affected)
Table 'Users'. Scan count 1, logical reads 31714, physical reads 0.

SQL
intersection

37

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Filters aren't necessarily more expensive.



Next query

```
SELECT Id
FROM dbo.Users
WHERE LastAccessDate > '7/1/2010'
ORDER BY LastAccessDate
```



39

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

The Execution Plan

1. Read all the Users records, only keeping the ones with LastAccessDate > '7/1/2010'
 2. As we work, write those records down somewhere, then return 'em
- But when do we sort?
 - As we write them to our staging, or
 - After we've written them down?



40

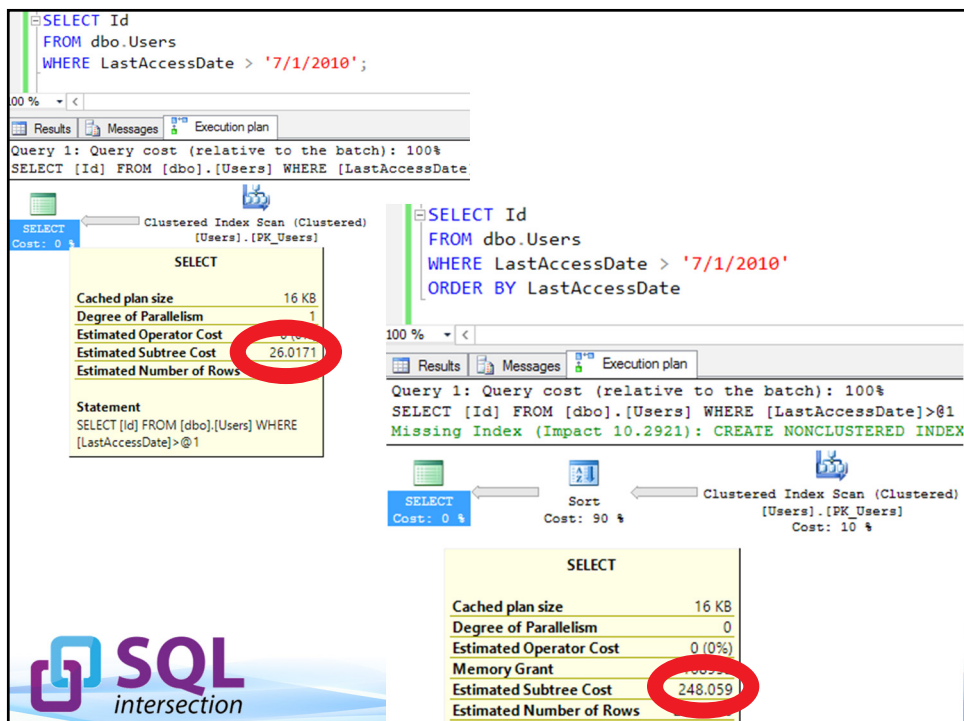
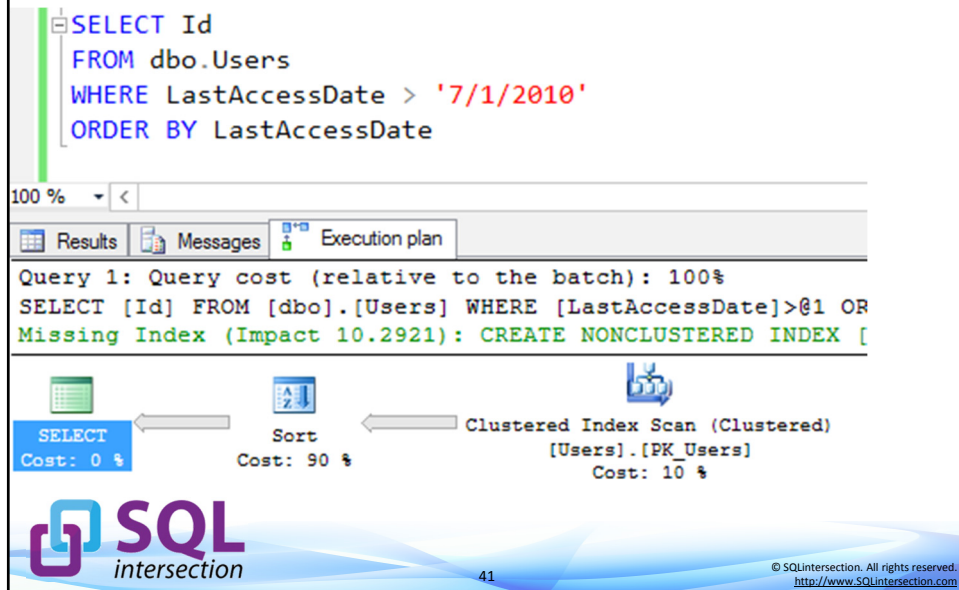
© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

SQLIntersection

Orlando, Florida – April 2014

www.SQLIntersection.com

SQL Server's Execution Plan



SQLIntersection

Orlando, Florida – April 2014

www.SQLIntersection.com

Parallelism might help.

```
SELECT Id
FROM dbo.Users
WHERE LastAccessDate > '7/1/2010'
ORDER BY LastAccessDate
```

Query 1: Query cost (relative to the batch): 100%

SELECT [Id] FROM [dbo].[Users] WHERE [LastAccessDate]>@1 ORDER BY [LastAccessDate]

Missing Index (Impact 15.7403): CREATE NONCLUSTERED INDEX [<Name of Missing Index>] ON [dbo].[Users] ([LastAccessDate])

Execution plan details:

- SELECT (Cost: 0 %)
- Parallelism (Gather Streams) (Cost: 10 %)
- Sort (Cost: 74 %)
- Clustered Index Scan (Clustered) ([Users].[PK_Users]) (Cost: 16 %)

SELECT details:

Cached plan size	24 KB
Degree of Parallelism	4
Estimated Operator Cost	150.492
Estimated Subtree Cost	150.492
Memory Grant	0 KB
Estimated Number of Rows	2278840

SQL intersection

43

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

```
/* Add in a sort, but don't parallelize: */
SELECT Id
FROM dbo.Users
WHERE LastAccessDate > '7/1/2010'
ORDER BY LastAccessDate OPTION (MAXDOP 1);

/* Go parallel by default: */
SELECT Id
FROM dbo.Users
WHERE LastAccessDate > '7/1/2010'
ORDER BY LastAccessDate;
```

Execution plan details:

- (2275167 row(s) affected)
- Table 'Users' Scan count 1, logical reads 31714, physical reads 0, (1 row(s) affected)
- (2275167 row(s) affected)
- Table 'Users' Scan count 5, logical reads 31936, physical reads 0, (1 row(s) affected)

SQL intersection

44

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Next query

```
SELECT *  
FROM dbo.Users  
WHERE LastAccessDate > '7/1/2010'  
ORDER BY LastAccessDate
```



45

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

That query sucks.

Why does it suck?

- Does it take more time to read?
- Does it take more time to write?
- Does it take more time to sort?
- Does it take more time to output?



47

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

```
SELECT Id
FROM dbo.Users
WHERE LastAccessDate > '7/1/2010'
ORDER BY LastAccessDate;

/* Compare adding all the fields: */
SELECT *
FROM dbo.Users
WHERE LastAccessDate > '7/1/2010'
ORDER BY LastAccessDate;
```

Results | Messages | Execution plan

(2275167 row(s) affected)
Table 'Users'. Scan count 5, logical reads 31936, physical reads 0, read-ahead reads 0, lob logical reads 0,
(1 row(s) affected)

(2275167 row(s) affected)
Table 'Users'. Scan count 5, logical reads 31904, physical reads 0, read-ahead reads 0, lob logical reads 69,
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, read-ahead reads 0, lob logical reads 0,
(1 row(s) affected)

Three red circles are drawn around the 'logical reads' values in the execution plan output: 31936 for the 'Users' table scan, 31904 for the 'Users' table scan in the second query, and 69 for the 'Worktable' scan in the second query.

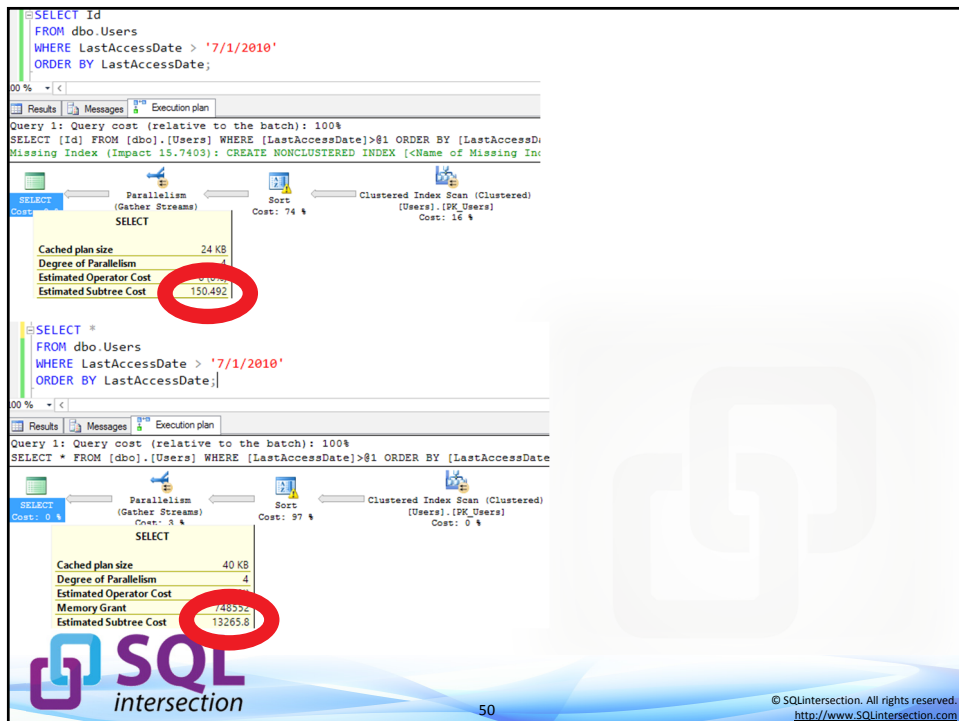
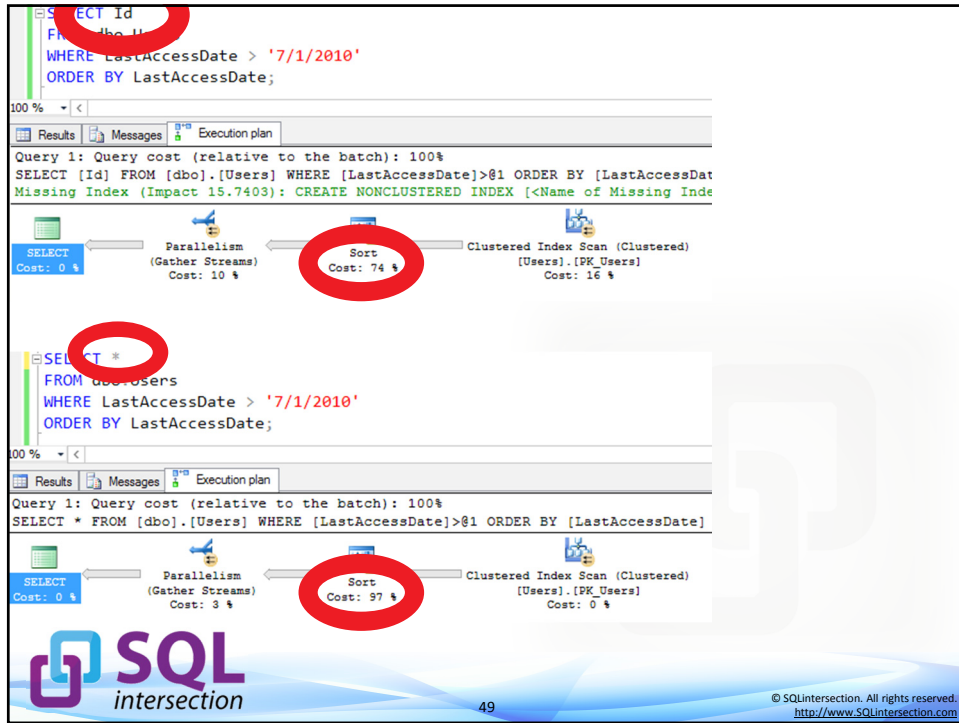
48

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

SQLIntersection

Orlando, Florida – April 2014

www.SQLIntersection.com



	SELECT ID	SELECT *
No ORDER BY	26	26
With ORDER BY	150	13,265

**Avoid sorting in the
database if you can,**

**but if you must do it, minimize
the fields involved.**

We need relief.

- Indexes: copies of the table
- Stored in the order we want
- Include the fields we want
- The index in your hands (black background):

```
CREATE INDEX  
IX_LastAccessDate_Id  
ON (LastAccessDate, Id)
```



53

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

dbo.Users - IX_LastAccessDate

LastAccessDate	Id	LastAccessDate	Id	LastAccessDate	Id	LastAccessDate	Id
7/31/08 12:00 AM	-1	7/15/09 8:53 AM	445	7/15/09 9:10 PM	200	8/11/09 7:17 PM	39
7/15/09 7:08 AM	22	7/15/09 8:58 AM	457	7/16/09 6:22 AM	678	8/12/09 2:54 PM	943
7/15/09 7:10 AM	33	7/15/09 9:17 AM	501	7/17/09 2:30 AM	131	8/13/09 4:26 PM	364
7/15/09 7:11 AM	40	7/15/09 9:28 AM	524	7/17/09 9:30 AM	297	8/15/09 5:03 PM	910
7/15/09 7:11 AM	41	7/15/09 9:30 AM	527	7/17/09 8:43 PM	998	8/17/09 8:42 AM	202
7/15/09 7:11 AM	44	7/15/09 9:58 AM	587	7/18/09 12:38 PM	394	8/17/09 10:11 AM	628
7/15/09 7:12 AM	52	7/15/09 10:00 AM	594	7/18/09 2:15 PM	924	8/17/09 10:33 AM	157
7/15/09 7:13 AM	64	7/15/09 10:02 AM	597	7/19/09 10:26 PM	336	8/17/09 4:24 PM	1006
7/15/09 7:13 AM	65	7/15/09 10:21 AM	618	7/20/09 1:06 PM	849	8/18/09 8:06 AM	511
7/15/09 7:14 AM	68	7/15/09 10:25 AM	347	7/21/09 7:22 AM	881	8/18/09 9:00 AM	262
7/15/09 7:15 AM	73	7/15/09 10:26 AM	623	7/23/09 11:53 AM	503	8/18/09 9:43 AM	210
7/15/09 7:17 AM	87	7/15/09 10:28 AM	629	7/23/09 12:56 PM	446	8/18/09 10:22 AM	673
7/15/09 7:18 AM	92	7/15/09 10:32 AM	638	7/24/09 12:15 AM	407	8/18/09 1:05 PM	959
7/15/09 7:20 AM	110	7/15/09 10:36 AM	642	7/24/09 12:08 PM	488	8/18/09 1:14 PM	643
7/15/09 7:26 AM	139	7/15/09 10:46 AM	661	7/24/09 3:17 PM	144	8/18/09 1:32 PM	525
7/15/09 7:36 AM	173	7/15/09 10:59 AM	686	7/26/09 8:48 AM	615	8/18/09 1:38 PM	159
7/15/09 7:38 AM	179	7/15/09 11:14 AM	722	7/27/09 2:27 PM	337	8/18/09 2:28 PM	690
7/15/09 7:46 AM	223	7/15/09 11:26 AM	756	7/28/09 3:17 AM	174	8/19/09 9:22 AM	750
7/15/09 7:57 AM	258	7/15/09 11:48 AM	803	7/28/09 12:34 PM	118	8/19/09 2:46 PM	887
7/15/09 7:58 AM	259	7/15/09 11:50 AM	681	7/28/09 9:28 PM	201	8/20/09 3:45 AM	749
7/15/09 8:05 AM	279	7/15/09 11:51 AM	808	7/28/09 10:53 PM	656	8/20/09 8:29 AM	268
7/15/09 8:06 AM	286	7/15/09 12:00 PM	822	7/29/09 12:06 PM	751	8/20/09 10:32 AM	901
7/15/09 8:12 AM	305	7/15/09 12:07 PM	841	7/30/09 12:26 PM	754	8/20/09 12:05 PM	752
7/15/09 8:13 AM	313	7/15/09 12:07 PM	842	7/31/09 6:36 PM	11	8/20/09 3:03 PM	272
7/15/09 8:14 AM	320	7/15/09 12:18 PM	878	8/1/09 12:09 PM	775	8/21/09 7:29 PM	487
7/15/09 8:16 AM	323	7/15/09 12:18 PM	779	8/1/09 7:29 PM	670	8/22/09 11:58 AM	580
7/15/09 8:25 AM	349	7/15/09 12:47 PM	973	8/1/09 11:33 PM	85	8/24/09 11:22 AM	56
7/15/09 8:26 AM	354	7/15/09 12:48 PM	819	8/3/09 10:13 PM	300	8/24/09 11:33 PM	889
7/15/09 8:27 AM	97	7/15/09 12:52 PM	987	8/5/09 12:06 PM	918	8/25/09 8:57 PM	150
7/15/09 8:27 AM	363	7/15/09 12:54 PM	992	8/6/09 5:46 AM	726	8/26/09 2:43 AM	851
7/15/09 8:33 AM	384	7/15/09 1:28 PM	931	8/6/09 10:42 AM	370	8/26/09 2:53 AM	117
7/15/09 8:37 AM	402	7/15/09 2:19 PM	728	8/6/09 12:37 PM	398	8/26/09 9:59 AM	546
7/15/09 8:47 AM	431	7/15/09 2:20 PM	217	8/9/09 5:28 AM	45	8/27/09 3:54 AM	239
7/15/09 8:52 AM	440	7/15/09 4:23 PM	975	8/10/09 12:25 AM	389	8/27/09 2:10 PM	271

From <http://StackOverflow.com>'s Creative Commons Data Dump



Next query, now with the index

```
SELECT Id
FROM dbo.Users
WHERE LastAccessDate > '7/1/2010'
ORDER BY LastAccessDate
```



55

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

The Execution Plan

1. Seek to 7/1/2010 on IX_LastAccessDate
2. Read them out in order



56

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

SQLIntersection

Orlando, Florida – April 2014

www.SQLIntersection.com

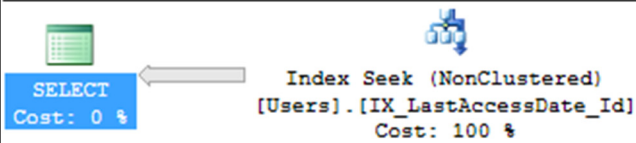
SQL Server's Execution Plan

```
SELECT Id
FROM dbo.Users
WHERE LastAccessDate > '7/1/2010'
ORDER BY LastAccessDate
```

100 %

Results Messages Execution plan

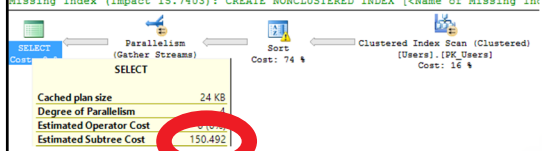
Query 1: Query cost (relative to the batch): 100%
SELECT [Id] FROM [dbo].[Users] WHERE [LastAccessDate]>@1 OR



57

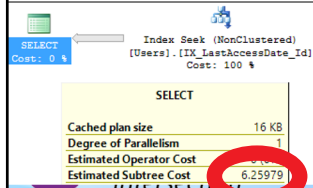
© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

```
SELECT Id
FROM dbo.Users
WHERE LastAccessDate > '7/1/2010'
ORDER BY LastAccessDate;
```



```
SELECT Id
FROM dbo.Users
WHERE LastAccessDate > '7/1/2010'
ORDER BY LastAccessDate
```

Query 1: Query cost (relative to the batch): 100%
SELECT [Id] FROM [dbo].[Users] WHERE [LastAccessDate]>@1



58

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>



SQLIntersection

Orlando, Florida – April 2014

www.SQLIntersection.com

```
SELECT Id
FROM dbo.Users
WHERE LastAccessDate > '7/1/2010'
ORDER BY LastAccessDate
```

(2275167 row(s) affected)
Table 'Users'. Scan count 5, logical reads 31936, physical reads 0,

```
SELECT Id
FROM dbo.Users
WHERE LastAccessDate > '7/1/2010'
ORDER BY LastAccessDate
```

(0 row(s) affected)
Table 'Users'. Scan count 1, logical reads 4, physical reads 0,

SQL intersection

59

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

**The index covers the fields
needed by the query,
so we call it a covering index.***

***But covering isn't really a special kind of index –
it's only covering when we're talking in terms of a query.**

SQL intersection

60

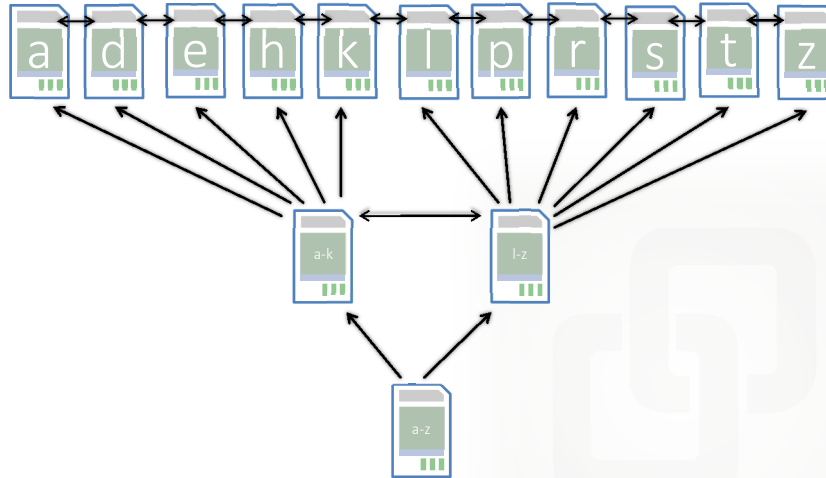
© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

SQLIntersection

Orlando, Florida – April 2014

www.SQLIntersection.com

Also, you're not holding the full index.



61

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Next query

```
SELECT Id, DisplayName, Age
FROM dbo.Users
WHERE LastAccessDate > '7/1/2010'
ORDER BY LastAccessDate
```



62

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>



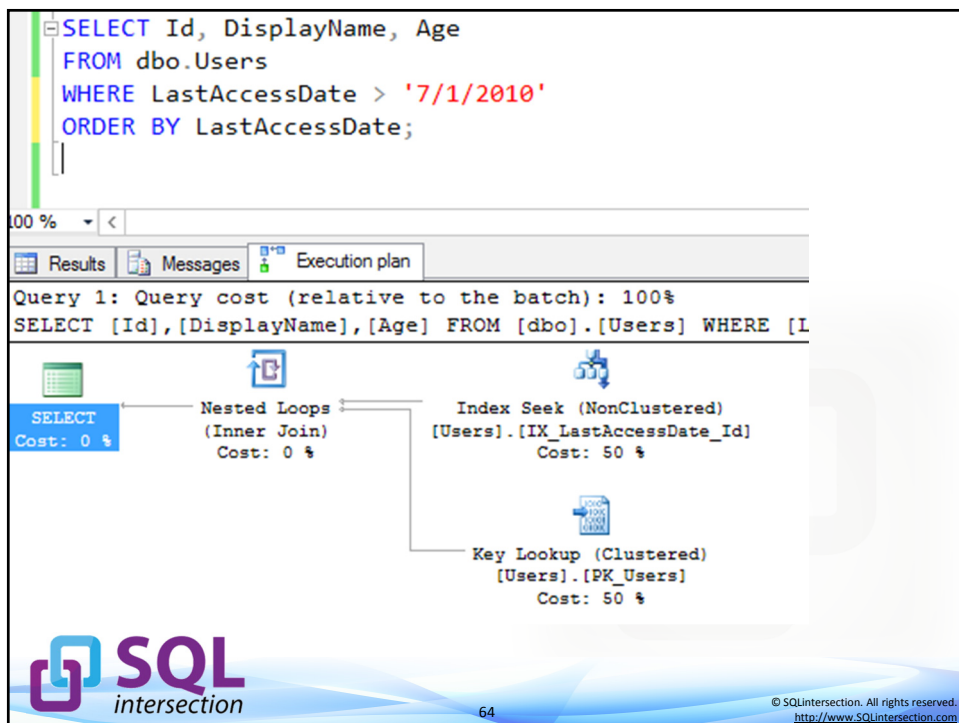
The Execution Plan

1. Seek to 7/1/2010 on IX_LastAccessDate
2. Read them out in order, make list
3. Look up the matching rows in the clustered index (table)



63

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>



SQLIntersection

Orlando, Florida – April 2014

www.SQLIntersection.com

relative to the batch):

me], [Age] FROM [dbo].[

Index Seek (NonClustered)

[Users].[IX_LastAccessDate]

Cost: 50 %

Key Lookup (Clustered)

[Users].[PK_Users]

Cost: 50 %

Physical Operation Key Lookup

Logical Operation Key Lookup

Estimated Execution Mode Row

Storage RowStore

Actual Number of Rows 0

Actual Number of Batches 0

Estimated Operator Cost 0.0032831 (50%)

Estimated I/O Cost 0.003125

Estimated CPU Cost 0.0001581

Estimated Subtree Cost 0.0032831

Estimated Number of Executions 1

Number of Executions 0

Estimated Number of Rows 1

Estimated Row Size 55 B

Actual Rebinds 0

Actual Rewinds 0

Ordered True

Node ID 3

Object

[StackOverflow].[dbo].[Users].[PK_Users]

Output List

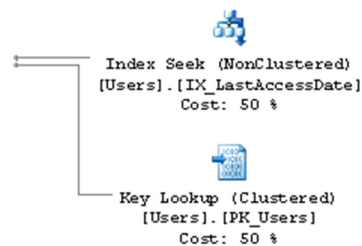
[StackOverflow].[dbo].[Users].Age, [StackOverflow].[dbo].[Users].DisplayName

Seek Predicates

Seek Keys[1]: Prefix: [StackOverflow].[dbo].[Users].Id = Scalar Operator([StackOverflow].[dbo].[Users].[Id])

Classic index tuning problem

- Key lookup is required when the index doesn't have all of the fields our query needs.
- Hover your mouse over the key lookup and look for the OUTPUT fields.
- Are they small and frequently used? Add them to the INCLUDE of the index, and see if the key lookup goes away.



SQLintersection

Orlando, Florida – April 2014

www.SQLintersection.com

- **CREATE INDEX**
IX_LastAccessDate_Id_DisplayName_Age
ON dbo.Users (LastAccessDate, Id, DisplayName, Age)
- **Or:**
- **CREATE INDEX**
IX_LastAccessDate_Id_DisplayName_Age
ON dbo.Users (LastAccessDate, Id)
INCLUDE (DisplayName, Age)

dbo.Users - IX_LastAccessDate_ID_DisplayName_Age

LastAccessDate	Id	DisplayName	Age	LastAccessDate	Id	DisplayName	Age
7/31/08 12:00 AM	-1	Community	1	8/18/09 10:22 AM	673		NULL
7/15/09 9:58 AM	587		NULL	8/18/09 1:14 PM	643	Sarcastic	30
7/15/09 10:00 AM	594	Louis Haußknecht	NULL	8/18/09 2:28 PM	690	msh	31
7/15/09 10:02 AM	597	elo80ka	29	8/19/09 9:22 AM	750	Cyphus	30
7/15/09 10:21 AM	618	Marien	30	8/20/09 3:45 AM	749		NULL
7/15/09 10:26 AM	623	Fionn	NULL	8/20/09 12:05 PM	752	Sean Taylor	27
7/15/09 10:28 AM	629	Zoomzoom83	26	8/22/09 11:58 AM	580	n00ki3	22
7/15/09 10:32 AM	638		NULL	8/26/09 2:43 AM	851	Tim Howland	41
7/15/09 10:36 AM	642	aherrick	23	8/26/09 9:59 AM	546	hitec	27
7/15/09 10:46 AM	661		NULL	8/28/09 7:51 PM	668	mtruesdell	34
7/15/09 10:59 AM	686	urini	38	9/1/09 1:39 AM	595	Salaryman	NULL
7/15/09 11:14 AM	722		NULL	9/6/09 9:24 PM	669	Tim Saunders	34
7/15/09 11:26 AM	756	suhair	29	9/8/09 2:56 AM	869	Malke	NULL
7/15/09 11:48 AM	803	Mike H	NULL	9/9/09 2:16 PM	644	spaetzel	27
7/15/09 11:50 AM	681	Foom	NULL	9/14/09 2:19 PM	707	17 of 26	38

Same query again

```
SELECT Id, DisplayName, Age
FROM dbo.Users
WHERE LastAccessDate > '7/1/2010'
ORDER BY LastAccessDate
```


SQLIntersection

Orlando, Florida – April 2014

www.SQLIntersection.com

The screenshot displays a SQL query in the query editor window:

```
SELECT Id, DisplayName, Age
FROM dbo.Users
WHERE LastAccessDate > '7/1/2010'
ORDER BY LastAccessDate;
```

Below the query, the 'Execution plan' tab is selected. The query cost is 100%. The execution plan shows a single step: 'Index Seek (NonClustered) [Users].[IX_LastAccessDate_Id_Displ...]' with a cost of 100%. A small icon representing the query is shown to the left of the execution plan step.

At the bottom of the slide, the SQL Intersection logo is visible on the left, the page number '69' is in the center, and the copyright notice '© SQLIntersection. All rights reserved. http://www.SQLIntersection.com' is on the right.

This screenshot is identical to the one above, showing the same SQL query and execution plan. However, a red circle is drawn around the small icon representing the query in the execution plan, highlighting it.

At the bottom of the slide, the SQL Intersection logo is visible on the left, the page number '70' is in the center, and the copyright notice '© SQLIntersection. All rights reserved. http://www.SQLIntersection.com' is on the right.

SQLIntersection

Orlando, Florida – April 2014

www.SQLIntersection.com

`SELECT Id, DisplayName, Age
FROM dbo.Users
WHERE LastAccessDate > '7/1/2010'
ORDER BY LastAccessDate;`

100 % <

Results Messages Execution plan

Query 1: Query cost (relative to the batch): 100
SELECT [Id],[DisplayName],[Age] FROM [dbo].[User

SELECT
Cost: 0 %

Index Seek (NonClustered)
(Users). [IX_LastAccessDate_Id_Displ...]
Cost: 100 %

Index Seek (NonClustered)
Scan a particular range of rows from a nonclustered index.

Physical Operation	Index Seek
Logical Operation	Index Seek
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Actual Number of Rows	2275167
Actual Number of Batches	0
Estimated Operator Cost	11.9546 (100%)
Estimated I/O Cost	9.45201
Estimated CPU Cost	2.50259
Estimated Subtree Cost	11.9546
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows	2274940

 SQL intersection

71

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

`SELECT Id, DisplayName, Age
FROM dbo.Users
WHERE LastAccessDate > '1/1/1800'
ORDER BY LastAccessDate;`

100 % <

Results Messages Execution plan

Query 1: Query cost (relative to the batch): 100
SELECT [Id],[DisplayName],[Age] FROM [dbo].[User

SELECT
Cost: 0 %

Index Seek (NonClustered)
[Users]. [IX_LastAccessDate_Id_Displ...]
Cost: 100 %

 SQL intersection

72

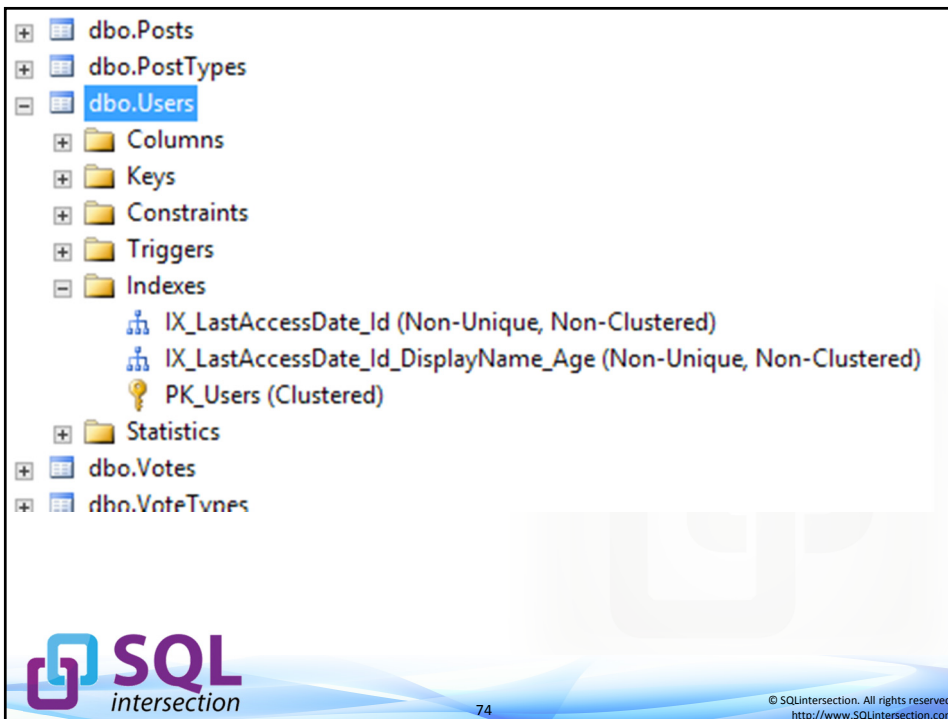
© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

“seek” != “awesome”



73

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>



SQLintersection

Orlando, Florida – April 2014

www.SQLintersection.com

More indexes mean

- Larger database
- Longer maintenance job times
(backups, DBCCs, index defrags)
- More work to do on inserts, updates, deletes
- More data to send to our replicas

Today, we'll teach you how to:

- Identify your app's problem
- Find the right solution
(which isn't always indexes)



75

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

SQLintersection
Session POSTCON01

Wait Stats: Focusing Your Tuning

Brent Ozar

Help@BrentOzar.com



SQLintersection

Orlando, Florida – April 2014

www.SQLintersection.com

Agenda

- The big picture of query processing
- Wait stats: what's the holdup?
- How to check a server's wait stats
- Common wait types



77

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

1st query: SELECT

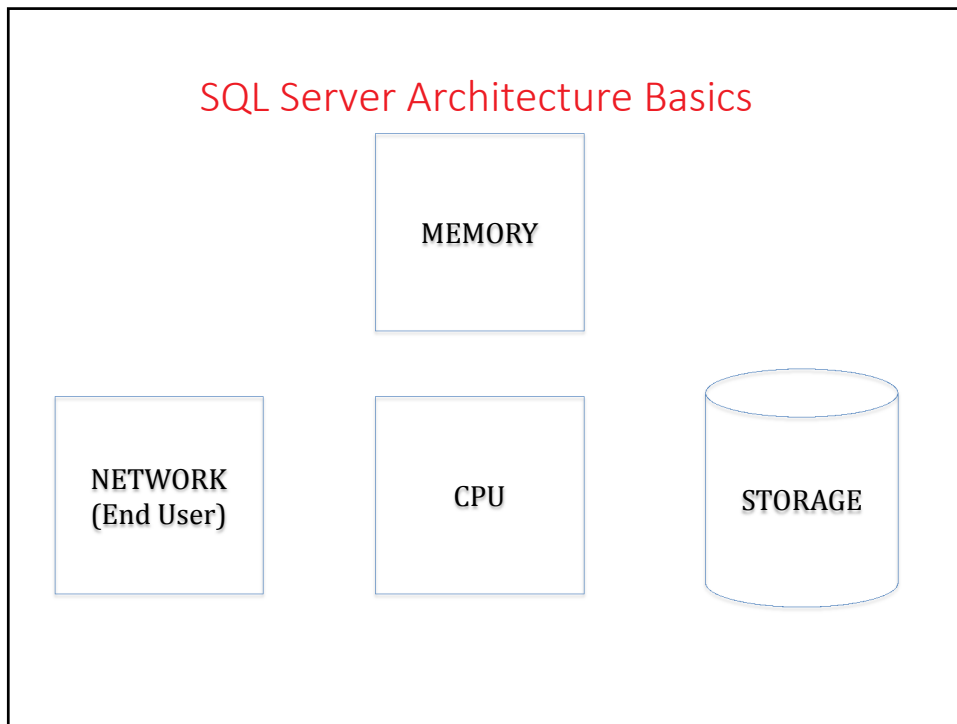


SQLintersection

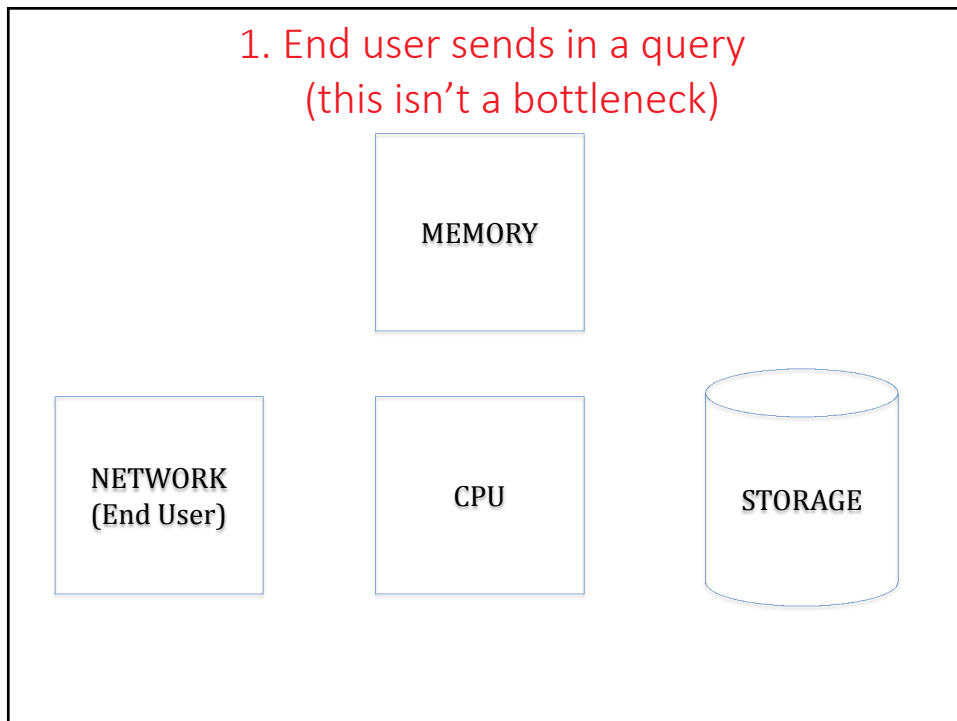
Orlando, Florida – April 2014

www.SQLintersection.com

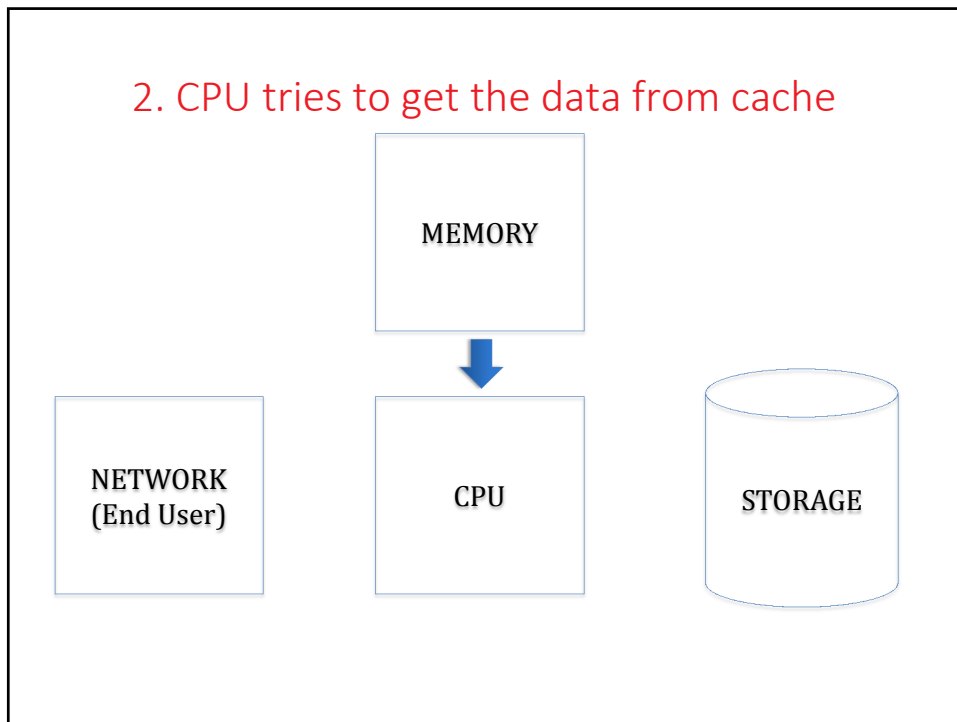
SQL Server Architecture Basics



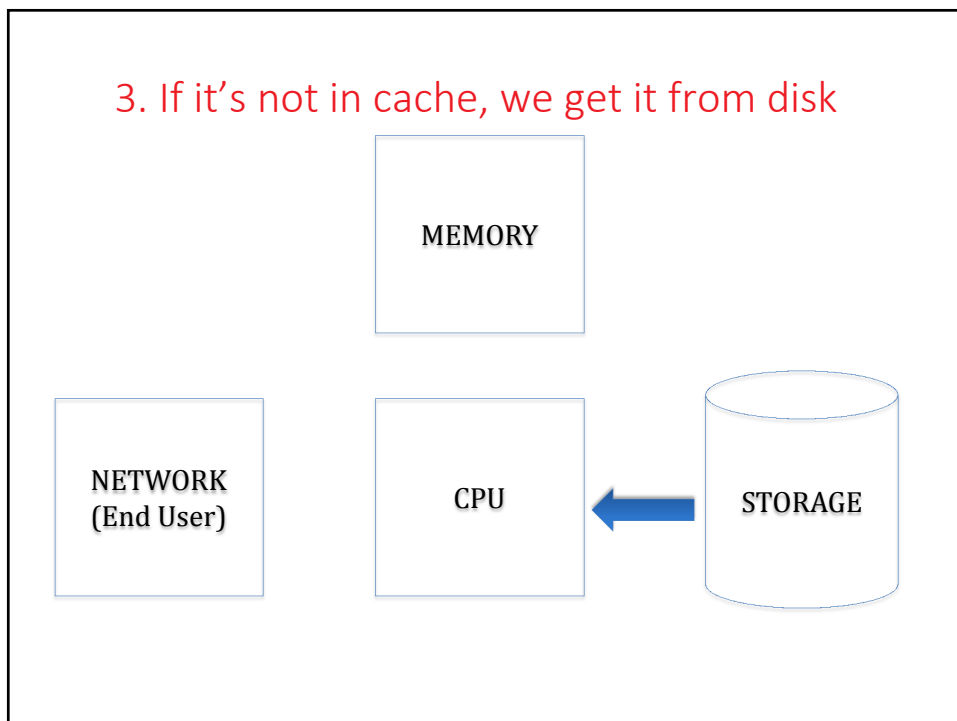
1. End user sends in a query
(this isn't a bottleneck)



2. CPU tries to get the data from cache



3. If it's not in cache, we get it from disk

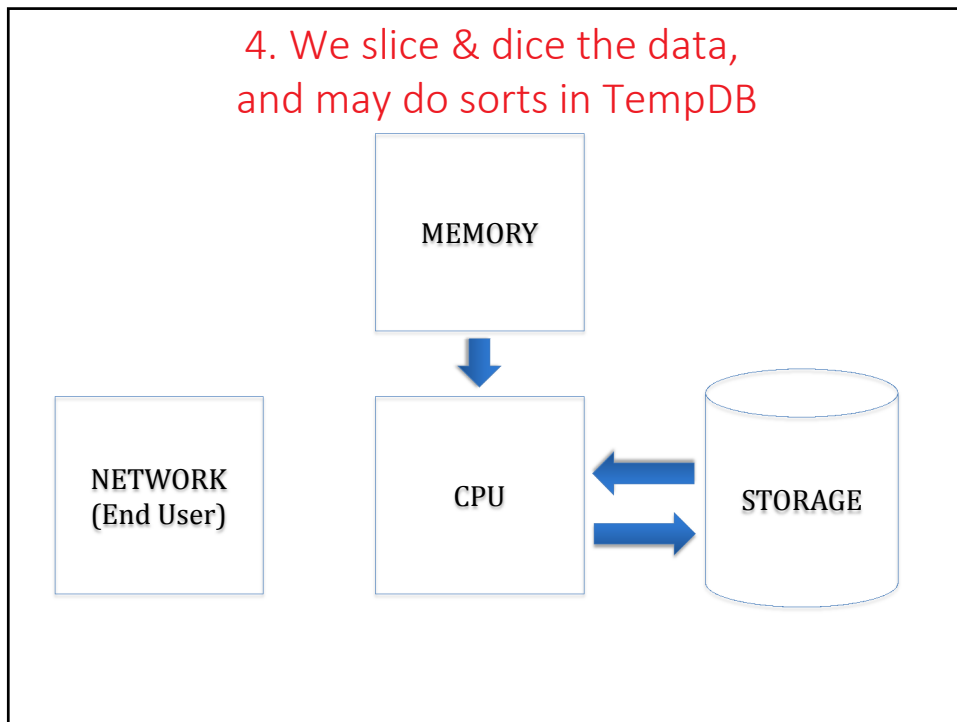


SQLintersection

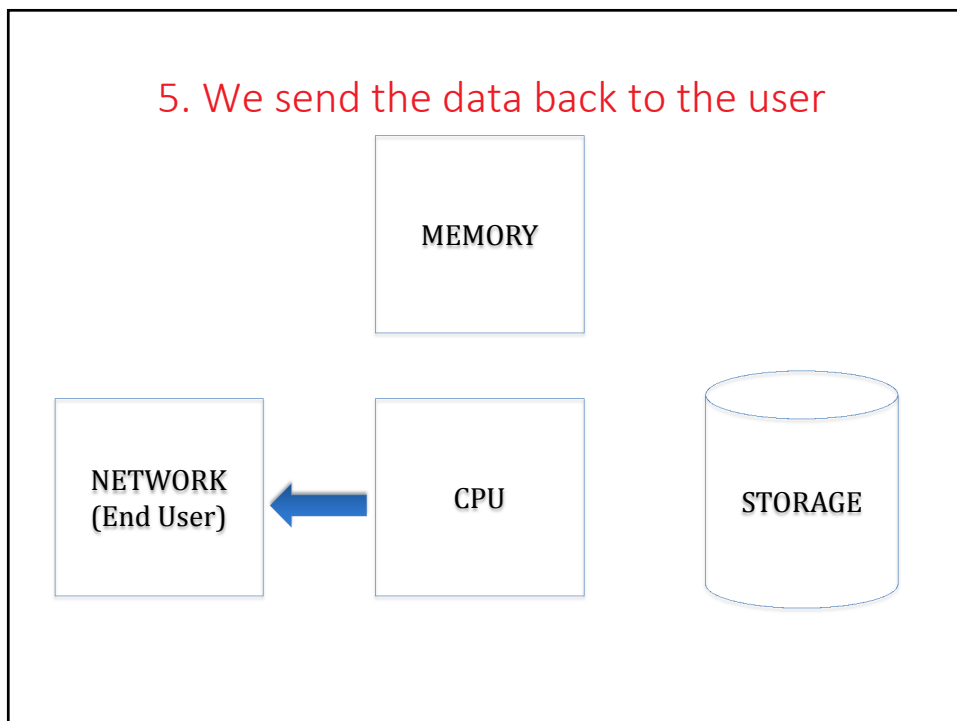
Orlando, Florida – April 2014

www.SQLintersection.com

4. We slice & dice the data,
and may do sorts in TempDB



5. We send the data back to the user



SQLintersection

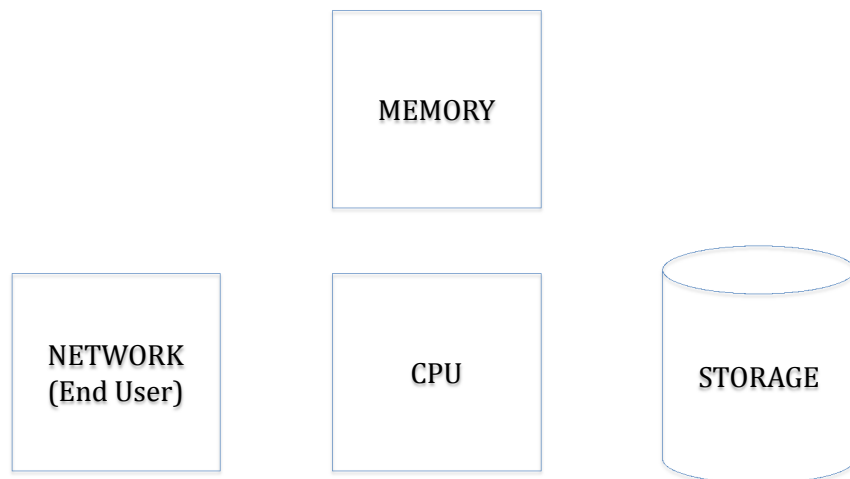
Orlando, Florida – April 2014

www.SQLintersection.com

2nd query: DELETE, UPDATE, INSERT (DUI) operations



1. User sends in the query
(again, not a bottleneck)

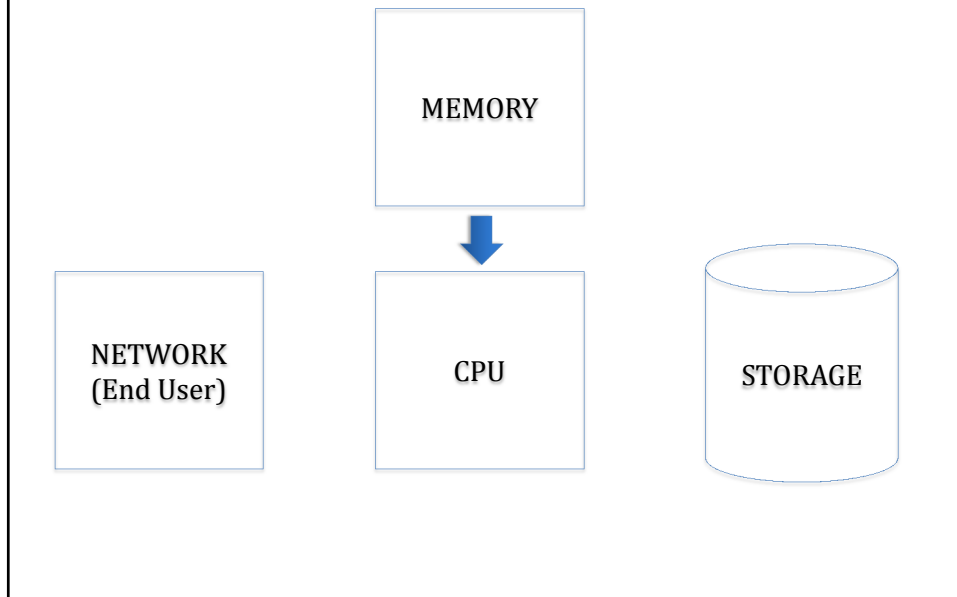


SQLintersection

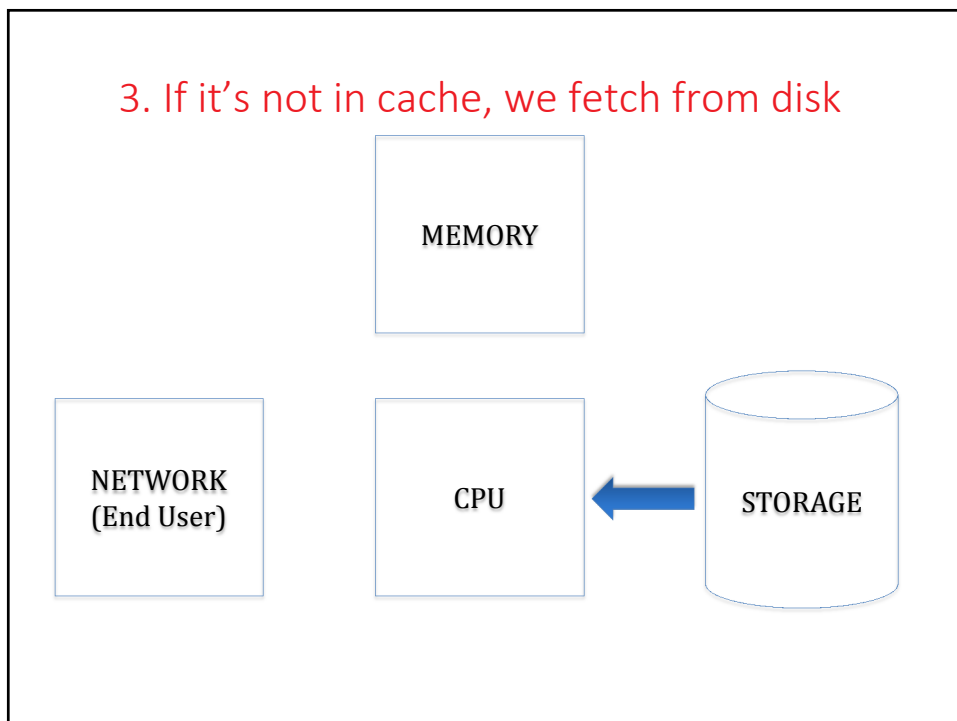
Orlando, Florida – April 2014

www.SQLintersection.com

2. We try to get related data into cache
(indexes, trigger tables)



3. If it's not in cache, we fetch from disk



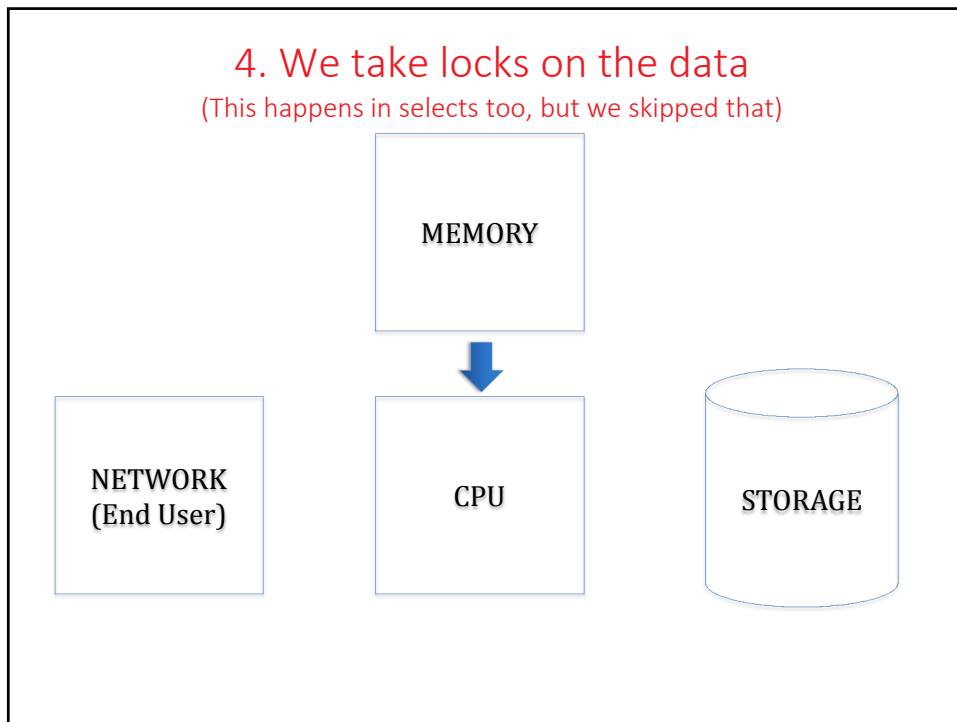
SQLIntersection

Orlando, Florida – April 2014

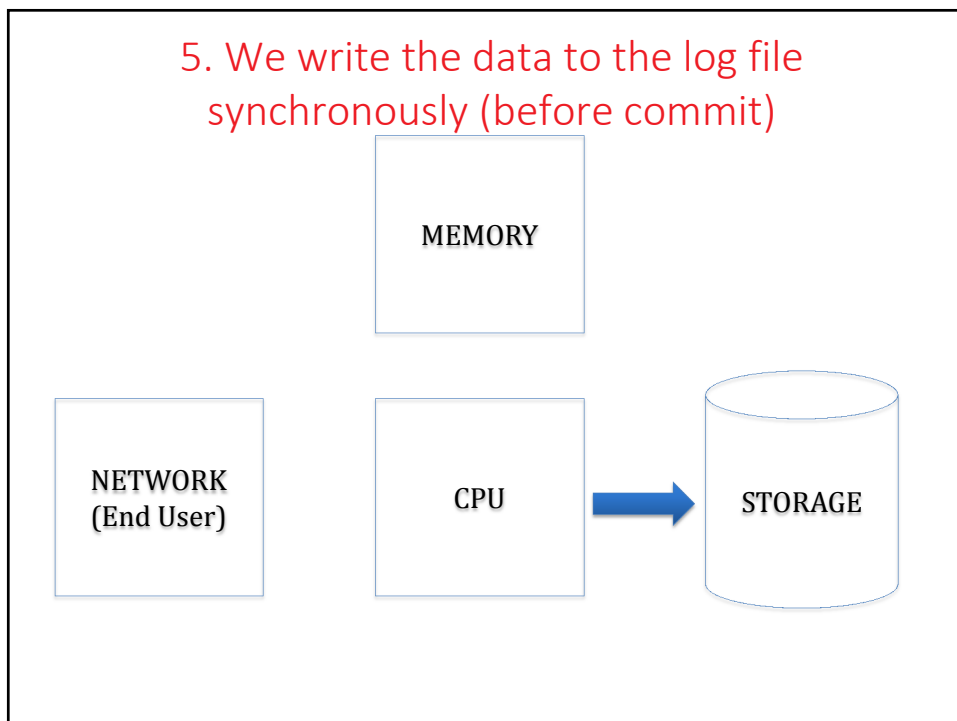
www.SQLIntersection.com

4. We take locks on the data

(This happens in selects too, but we skipped that)



5. We write the data to the log file synchronously (before commit)

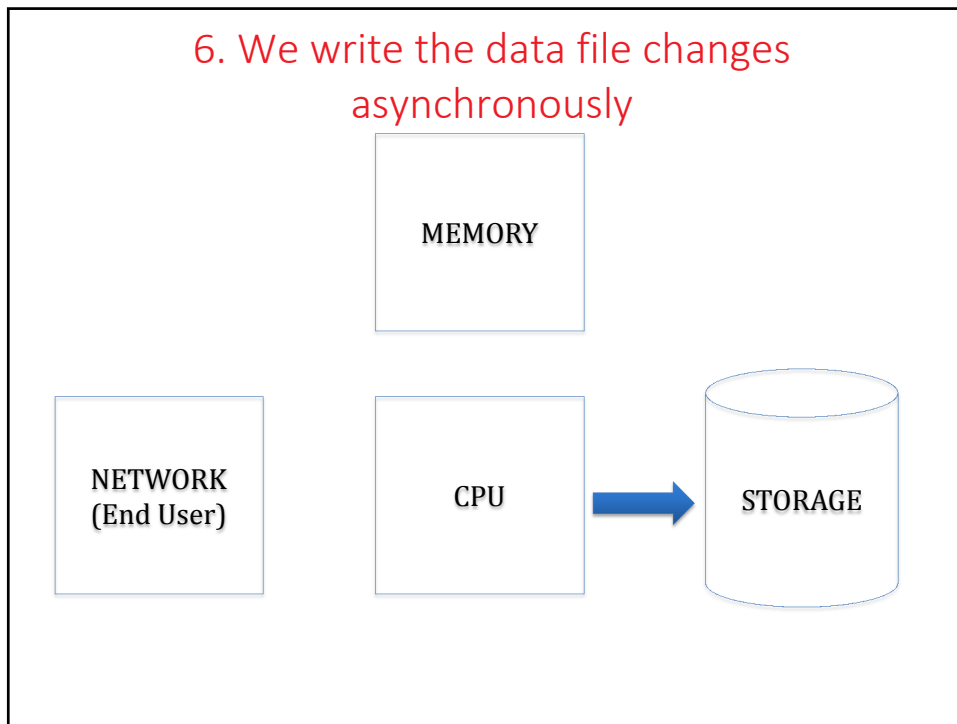


SQLintersection

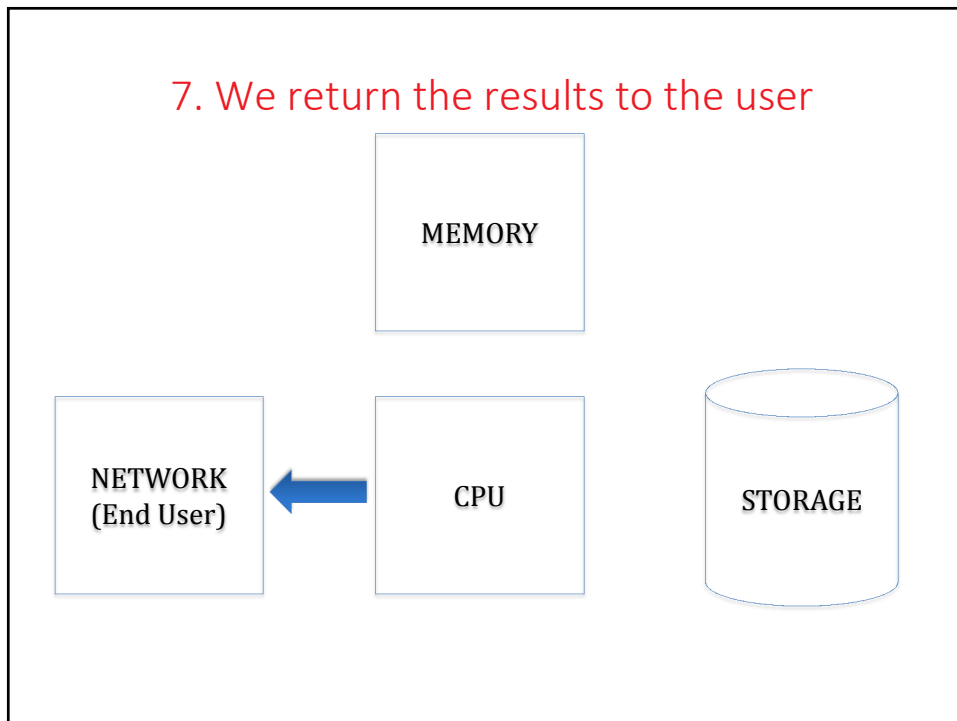
Orlando, Florida – April 2014

www.SQLintersection.com

6. We write the data file changes
asynchronously



7. We return the results to the user



**Wait stats:
how SQL Server
tracks all these arrows.**

How SQL Server Schedules CPU

What's Running Now

What's Waiting (Queue)

How SQL Server Schedules CPU

What's Running Now

- **SELECT * FROM
dbo.Restaurants
(By Brent)**

What's Waiting (Queue)

How SQL Server Schedules CPU

What's Running Now

- **SELECT * FROM
dbo.Restaurants
(By Brent)**

What's Waiting (Queue)

- **SELECT * FROM
dbo.Tattoos
(By Jeremiah)**
- **SELECT * FROM
dbo.Rabbits
(By Kendra)**

SQLintersection

Orlando, Florida – April 2014

www.SQLintersection.com

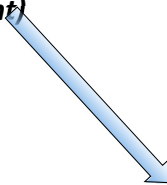
How SQL Server Schedules CPU

What's Running Now

- **SELECT * FROM
dbo.Restaurants
(By Brent)**

What's Waiting (Queue)

- **SELECT * FROM
dbo.Tattoos
(By Jeremiah)**
- **SELECT * FROM
dbo.Rabbits
(By Kendra)**



How SQL Server Schedules CPU

What's Running Now

What's Waiting (Queue)



- **SELECT * FROM
dbo.Tattoos
(By Jeremiah)**
- **SELECT * FROM
dbo.Rabbits
(By Kendra)**
- **SELECT * FROM
dbo.Restaurants
(By Brent)**

SQLintersection

Orlando, Florida – April 2014

www.SQLintersection.com

That's the Simple Version

What's Running Now

- **SELECT * FROM
dbo.Tattoos
(By Jeremiah)**

What's Waiting (Queue)

- **SELECT * FROM
dbo.Rabbits
(By Kendra)**
- **SELECT * FROM
dbo.Restaurants
(By Brent)**

Slightly More Complicated

Running

- **SELECT *
FROM dbo.Tattoos
(By Jeremiah)**

Runnable

- **SELECT *
FROM dbo.Rabbits
(By Kendra)**

Waiting (Queued)

- **SELECT *
FROM
dbo.Restaurants
(By Brent)**
- **SELECT *
FROM dbo.Races
(By Jes)**



SQL Server Waits For:

- **Resources:**
CPU, memory, storage, network, latches, locks
(the same stuff we showed in the life of a query!)
- **Stuff Outside SQL Server (Preemptive):**
COM, OLEDB, CLR
- **System Tasks:**
Lazywriter, trace, full text search



101

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Tracking the bottleneck

- **Old way:**
 - Measure every choke point (CPU % busy, disk queue lengths, memory usage, etc).
 - Compare our metrics against Microsoft's guidelines
 - Improve anything that seemed to be slow (and hope queries get faster)
- **New way:**
 - Ask SQL Server, "What have you been waiting on?"



102

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

How to check wait stats



Option #1: over time BrentOzar.com/go/waitsnow

**BRENT OZAR**
UNLIMITED

HOME PROBLEMS
We Solve SERVICES
We Provide FIRST AID

First Responder Kit
[Get Top Resource-Consuming Queries](#)
[Triage Wait Stats in SQL Server](#)
[Log Activity with sp_whoisactive](#)

Triage Wait Stats in SQL Server

My SQL Server is online, but it's incredibly slow.

What have my queries in SQL Server been waiting on? And more importantly, what are they waiting on *now*?

This query helps you find out just what you've been waiting on. First it returns the overall waits on the instance since SQL Server startup (or since the last time the wait data was manually cleared). Then it takes a 30 second sample of what your system is waiting on *right now*.

```
/* Copyright Brent Ozar Unlimited, 2013.
SQL Server Wait Information from sys.dm_os_wait_stats

Community references for waits:
sys.dm_os_wait_stats Books Online: http://msdn.microsoft.com/en-us/library/ms179984(v=sql.105).aspx
Microsoft PSS Wait Type Repository:
http://blogs.msdn.com/b/psssql/archive/2009/11/03/the-sql-server-wait-type-repository.aspx
Service Broker Wait Types Explained:
http://blogs.msdn.com/b/sql_service_broker/archive/2008/12/01/service-broker-wait-types.aspx
Technet Magazine, "SQL Server: SQL Server Delays Demystified"
http://technet.microsoft.com/en-us/magazine/hh781189.aspx
*/

/*****
Let's build a list of waits we can safely ignore.
*****/
IF OBJECT_ID('tempdb..#ignorable_waits') IS NOT NULL
```

What the script does

1. Instantly returns one set of results with cumulative updates since the last server restart, or since waits were cleared manually
2. Then:
 - Waits 30 seconds (don't panic)
 - Returns another set of results with wait stats over that last 30 seconds



105

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

The results: two cryptic lists of waits

wait_type	sum_wait_time_ms	pct_wait_time	sum_waiting_tasks	avg_wait_time_ms	sample_time
1 LCK_M_X	45946	69.3	3	15315.3	2013-02-27 13:08:33.427
2 MSOL_XP	4451	6.7	109	40.8	2013-02-27 13:08:33.427
3 ASYNC_NETWORK_IO	2790	4.2	1914	1.5	2013-02-27 13:08:33.427
4 PREEMPTIVE_OS_WAITFORSINGLEOBJECT	2712	4.1	613	4.4	2013-02-27 13:08:33.427
5 LCK_M_S	1151	1.7	15	76.7	2013-02-27 13:08:33.427
6 PREEMPTIVE_OS_FLUSHFILEBUFFERS	1120	1.7	38	29.5	2013-02-27 13:08:33.427
7 PREEMPTIVE_OS_AUTHENTICATIONOPS	830	1.3	6345	0.1	2013-02-27 13:08:33.427
8 PREEMPTIVE_OS_AUTHORIZATIONOPS	810	1.2	6345	0.1	2013-02-27 13:08:33.427

Second Sample Time	Sample Duration in Seconds	wait_type	Wait Time (Seconds)	Number of Waits	Avg ms Per Wait
1 2013-02-27 13:09:03.517	30	ASYNC_NETWORK_IO	0.0	19	0.0
2 2013-02-27 13:09:03.517	30	PREEMPTIVE_OS_AUTHENTICATIONOPS	0.0	152	0.1
3 2013-02-27 13:09:03.517	30	PREEMPTIVE_OS_AUTHORIZATIONOPS	0.0	25	0.1
4 2013-02-27 13:09:03.517	30	PREEMPTIVE_OS_DECRYPTMESSAGE	0.0	25	0.0
5 2013-02-27 13:09:03.517	30	PREEMPTIVE_OS_DELETESECURITYCONTEXT	0.0	26	0.2
6 2013-02-27 13:09:03.517	30	PREEMPTIVE_OS_DISCONNECTNAMEDPIPE	0.0	26	0.0
7 2013-02-27 13:09:03.517	30	PREEMPTIVE_OS_QUERYCONTEXTATTRIBUTES	0.0	26	0.0
8 2013-02-27 13:09:03.517	30	PREEMPTIVE_OS_REVERTTOSELF	0.0	26	0.0
9 2013-02-27 13:09:03.517	30	PREEMPTIVE_OS_WRITEFILE	0.0	1	3.0
10 2013-02-27 13:09:03.517	30	SOS_SCHEDULER_YIELD	0.0	5	0.0

- Top: over time. Bottom: last 30 seconds.



106

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>



Option #2: now, 5-second sample:
BrentOzar.com/askbrent

The screenshot shows the Brent Ozar Unlimited website. The header includes the logo and navigation links: HOME, PROBLEMS (We Solve), SERVICES (We Provide), and FIRST AID. Below the header is a 'Blog' section with a 'Subscribe' button. The main content area features a blog post titled 'SQL Server DBA Scripts: How to Find Slow SQL Server Queries' by Brent Ozar, dated September 9, 2010, with 40 comments. The post text mentions that SQL Server database administrators need to be able to quickly find out what queries and stored procedures are running slow, and that there's a much better tool, which is completely free. It also mentions a five-minute tutorial video explaining how to use sp_WhoIsActive from Adam Machanic. Below the text is a video player showing a man speaking. To the right of the post is a 'Need Help?' section with contact information: (773) 420-9626 and a promise that it won't hurt. Below that is a 'Popular Topics' section with links to various SQL Server topics.

SQL Server database administrators need to be able to quickly find out what queries and stored procedures are running slow. Microsoft includes sp_who and sp_who2 in SQL Server 2005 and 2008, but there's a much better tool, and it's completely free.

In this five minute tutorial video, I explain how to use sp_WhoIsActive from Adam Machanic ([Blog](#) – [@AdamMachanic](#)). Sorry about the audio – I've ordered a new microphone and it hasn't arrived yet.

How to Find Slow SQL Server Queries

login_name	cpu
per... SQLNEXUS\svcSQL	43.
per... SQLNEXUS\svcSQL	45.

Common wait types

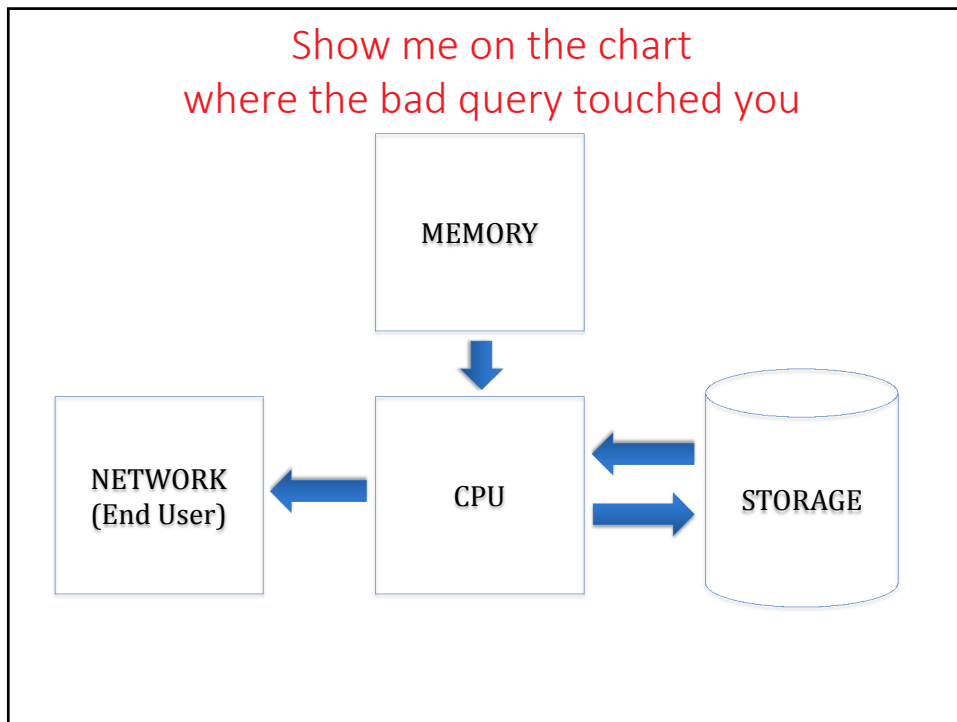


SQLintersection

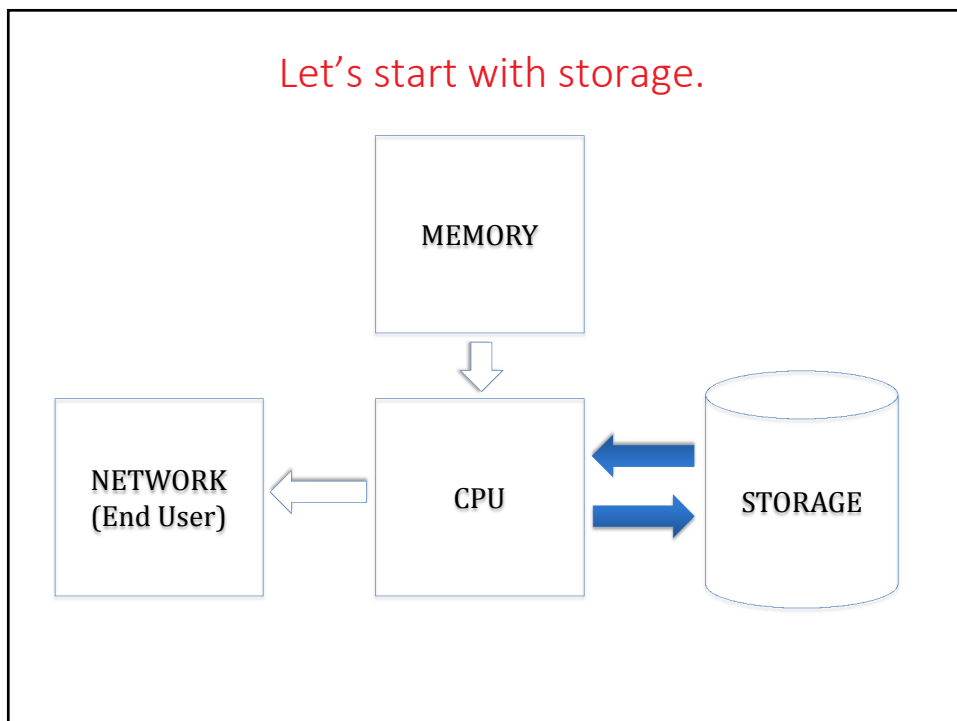
Orlando, Florida – April 2014

www.SQLintersection.com

Show me on the chart
where the bad query touched you



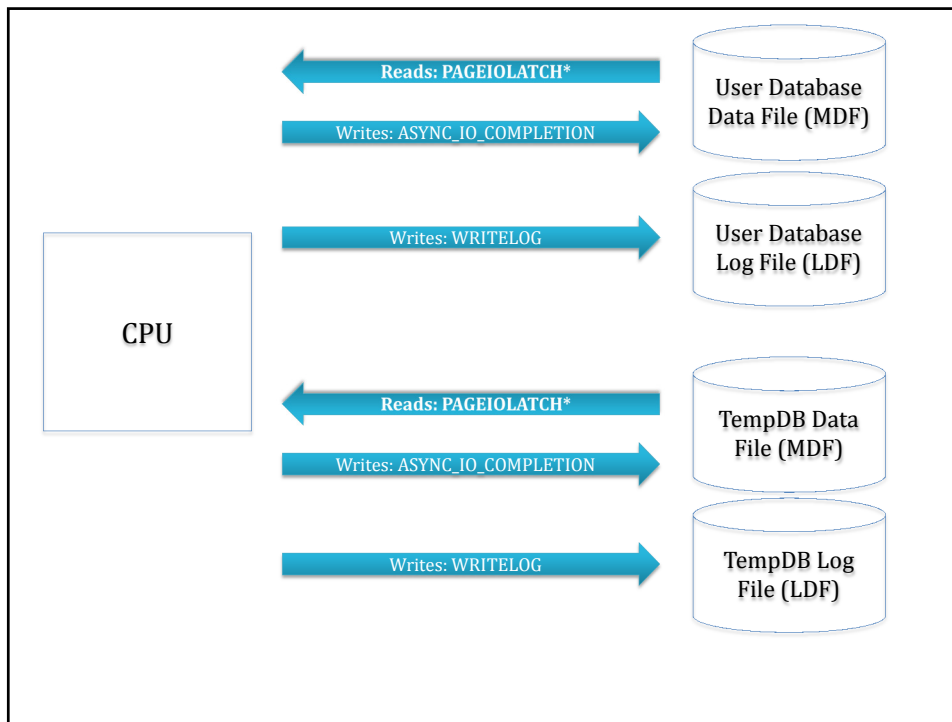
Let's start with storage.



SQLintersection

Orlando, Florida – April 2014

www.SQLintersection.com



Perfmon counters for more info

- Performance Monitor counters:
 - Physical Disk: Avg Sec/Read
 - Physical Disk: Avg Sec/Write
- Reported in whole seconds – use 3 decimal places for MS
- Reported at the whole-drive level
- SAN equivalent term: latency
- Good for both physical and virtual servers
- Related counters: Physical Disk: Reads/sec, Writes/sec (shows how much we're asking storage to work)
- The more we ask storage to work, the slower it'll get.

File-by-file details

- To get at the database or file level,
query sys.dm_io_virtual_file_stats
- Great query: BrentOzar.com/go/dmvs
- Easier way: sp_AskBrent @ExpertMode = 1

```
-- Calculates average stalls per read, per write, and per total input/output for each database file.  
SELECT DB_NAME(fs.database_id) AS [Database Name], mf.physical_name, io_stall_read_ms, num_of_reads,  
CAST(io_stall_read_ms/(1.0 + num_of_reads) AS NUMERIC(10,1)) AS [avg_read_stall_ms],io_stall_write_ms,  
num_of_writes,CAST(io_stall_write_ms/(1.0+num_of_writes) AS NUMERIC(10,1)) AS [avg_write_stall_ms],  
io_stall_read_ms + io_stall_write_ms AS [io_stalls], num_of_reads + num_of_writes AS [total_io],  
CAST((io_stall_read_ms + io_stall_write_ms)/(1.0 + num_of_reads + num_of_writes) AS NUMERIC(10,1))  
AS [avg_io_stall_ms]  
FROM sys.dm_io_virtual_file_stats(null,null) AS fs  
INNER JOIN sys.master_files AS mf  
ON fs.database_id = mf.database_id  
AND fs.[file_id] = mf.[file_id]  
ORDER BY avg_io_stall_ms DESC OPTION (RECOMPILE);
```



113

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Related memory counters

- Perfmon counter –
SQL Server Buffer Manager: Page Life Expectancy
- Number of seconds SQL can keep data cached
after reading it from storage
- Longer = better (should be at least 300, aka 5min)
- If PLE is low, then storage may be being forced to work too hard. Better
solution – add memory.



114

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Fixing PAGEIOLATCH waits

- Make storage faster
- Add more memory
- Tune indexes
- Tune queries



115

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Fixing WRITELOG waits

- Write less data:
 - Move processing tables into their own DB
 - Question code that writes data, like triggers
 - Move low-value databases to other servers
- Make storage faster



116

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Fixing ASYNC_IO_COMPLETION

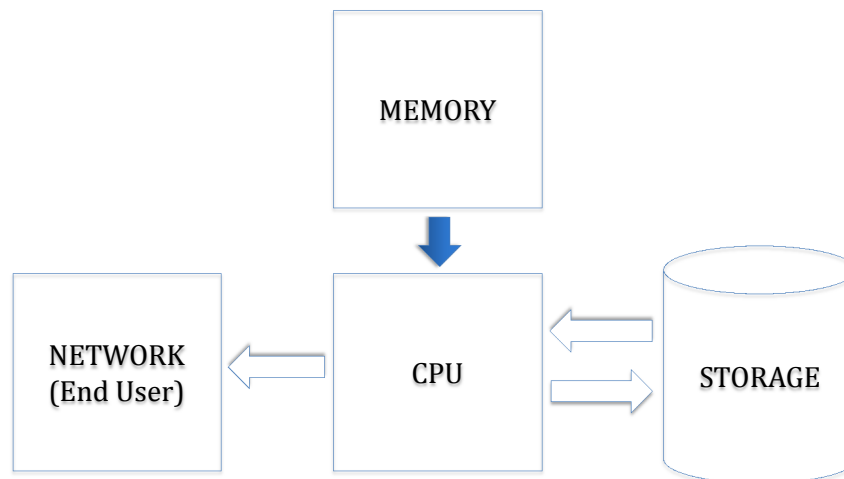
- Make sure your users actually care first.



117

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Next, memory.



How memory shows up as a wait

- Table: `dbo.CustomerConfigValues`
- Has no indexes. None.
- Has 50,000 rows for configurations for all of our customers
- Our app constantly runs this query:

```
SELECT *  
FROM dbo.Customers  
WHERE CustomerID = 123  
AND ConfigParameter = 'LoginTimeout'
```
- Result: we're constantly scanning the table.



119

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

PAGELATCH* waits

- Before SQL Server can read a page, it needs to take a quick look to see if anyone has a lock on that page.
- Latch: like a lightweight lock for a quick look.
- Normally, latches wouldn't show up as a significant overall wait. Each latch is blazing fast.
- However, if we're constantly reading pages, we end up taking a lot of latches.



120

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Fixing PAGELATCH* waits

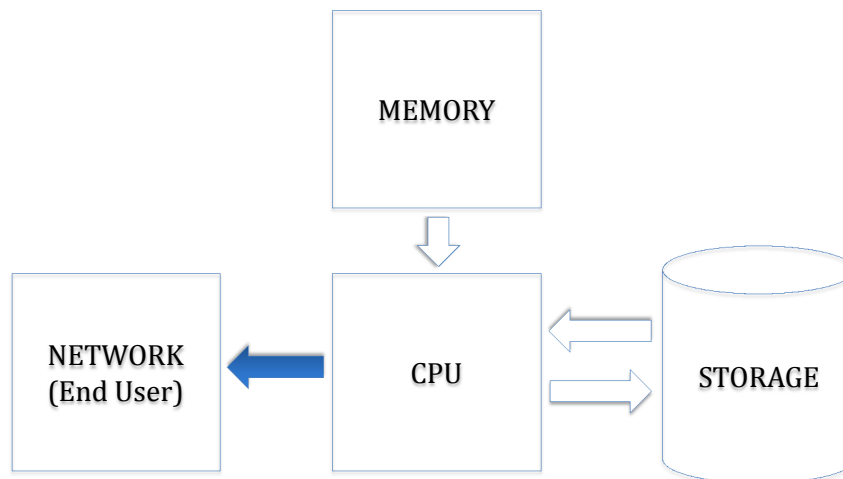
- ~~Make memory faster~~
- Find the queries doing tons of reads, and put the right index on it.



121

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Next up, network.



When does SQL wait on network?

- Slow networks (WANs)
- Small shares of network (25 chatty VMs on one host)
- Slow client computers
- Slow applications that process data row-by-row
- Applications that ask for too much data



123

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

When does SQL wait on network?

- Slow networks (WANs)
- Small shares of network (25 chatty VMs on one host)
- Slow client computers
- Slow applications that process data row-by-row
- Applications that ask for too much data
- All of this looks the same to SQL Server:
ASYNC_NETWORK_IO waits show up.
- Good news: it's not our problem!



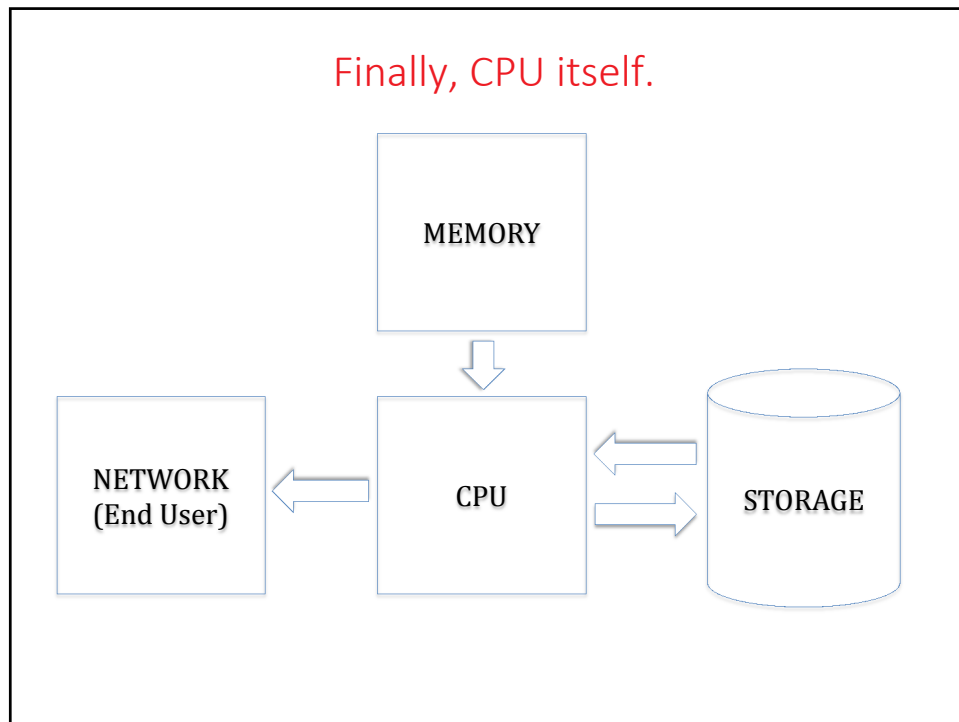
124

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

SQLIntersection

Orlando, Florida – April 2014

www.SQLIntersection.com



CPU is different.

- When queries are runnable, but not running, that means we're waiting on CPU.
- Surfaces as SIGNAL waits, which are calculated differently and are beyond what we'll show here.
- Instead, Perfmon counters are the easy button:
 - Processor/ % Used
Measure all cores individually, not just total
 - System/Processor Queue Length

SQLIntersection

Orlando, Florida – April 2014

www.SQLIntersection.com

Recap

- The life of a query: selects and insert/update/delete
- Wait stats: SQL Server tracks what we're waiting on
- How to check a server's wait stats
- Defined common wait types



127

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

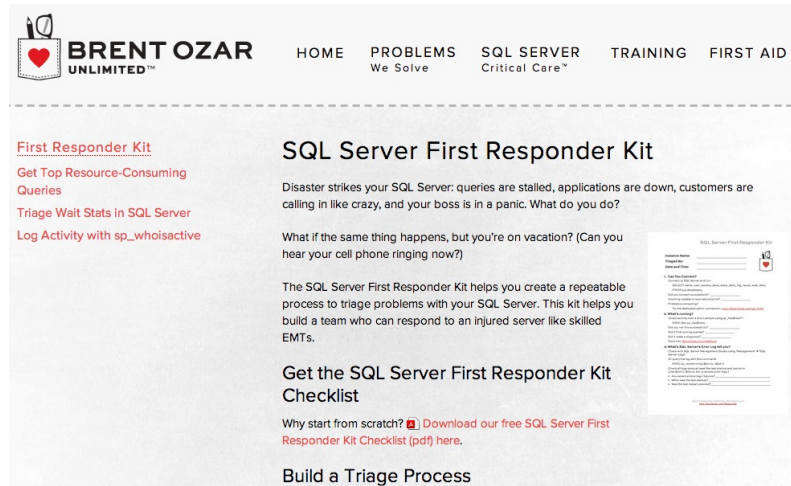
SQLIntersection
POSTCON01

What's Happening in my SQL Server?

Jeremiah Peschka
jeremiah@brentozar.com



BrentOzar.com/responder



129

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

BrentOzar.com/responder

Tools for Triaging SQL Server

FIND SLOW QUERIES WITH SP_ASKBRENT™, SP_WHIOISACTIVE, AND MORE

Need to find out what queries are killing your server right now? Here's some of our favorites:

- **sp_AskBrent™** takes a 10-second sample of your SQL Server's vital statistics and reports back what the wait stat bottlenecks are, which queries ran the most often, and more.
- **Adam Machanic's sp_whoisactive stored procedure** also shows what's running, and what it's waiting on. For longer-term diagnostics, here's how to **run sp_whoisactive in a loop and store data to a table for later review**.
- **ClearTrace** is a slice-and-dice BI-style interface for your Profiler trace files.
- **Extended Events** can help you catch failing queries that error out.

SP_SERVER_DIAGNOSTICS

- Read **Bob Dorr's post about SQL Server 2012's sp_server_diagnostics**
- Check out the **sp_server_diagnostics Books Online page**

THE DEDICATED ADMIN CONNECTION



130

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>



SQL Server is helpful

- **Every step of the way, an incredible amount of information is collected.**
 - Wait stats.
 - Hardware metrics (as SQL Server sees it).
 - Query execution statistics.
- **We can use this information to diagnose SQL Server performance problems.**



131

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

We're going to cover 4 tools

- `sp_WhoIsActive`
- `sp_AskBrent®`
- `sp_BlitzCache™`
- SQL Server Extended Events



132

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

sp_WholsActive



sp_WholsActive
answers the question:
“What stinks... *right*
now?”

dd hh:mm:ss.mss	session_id	sql_text	login_name	wait_info	CPU	tempdb_allocations	tempdb_current	blocking_session_id
00:00:00.28146	55	sp_WhoIsActive @get_task_info = 1	sa	NULL	0	0	0	NULL
00:00:00.01340	103	UPDATE dbo.ORDERS SET O.CARRIER_ID = @d	sa	(40ms)WRTLELOG	1	0	0	NULL
00:00:00.01320	93	UPDATE dbo.ORDERS SET O.CARRIER_ID = @d	sa	(19ms)WRTLELOG	8	0	0	NULL
00:00:00.01270	118	DELETE dbo.NEW_ORDER WHERE NEW_ORDER	sa	(8ms)WRTLELOG	4	0	0	NULL
00:00:00.01226	59	UPDATE dbo.CUSTOMER SET C.BALANCE = @US	sa	(5ms)WRTLELOG	6	0	0	NULL
00:00:00.01150	99	INSERT INTO BackOrderResults (CheckID, Priority, Fi	WIN-MP3KEBL8ASC\Administrator	NULL	1,107	36	36	NULL
00:00:00.01143	63	UPDATE dbo.ORDERS SET O.CARRIER_ID = @d	sa	(13ms)WRTLELOG	2	0	0	NULL
00:00:00.01113	80	DELETE dbo.NEW_ORDER WHERE NEW_ORDER	sa	(19ms)WRTLELOG	2	0	0	NULL
00:00:00.01103	51	UPDATE dbo.ORDERS SET O.CARRIER_ID = @d	sa	(19ms)WRTLELOG	6	0	0	NULL
00:00:00.01056	76	UPDATE dbo.CUSTOMER SET C.BALANCE = @US	sa	(18ms)WRTLELOG	4	0	0	NULL
00:00:00.01033	111	UPDATE dbo.CUSTOMER SET C.BALANCE = @US	sa	(16ms)WRTLELOG	3	0	0	NULL
00:00:00.00970	119	UPDATE dbo.ORDERS SET O.CARRIER_ID = @d	sa	(16ms)WRTLELOG	0	0	0	NULL
00:00:00.00963	109	UPDATE dbo.ORDERS SET O.CARRIER_ID = @d	sa	(8ms)WRTLELOG	1	0	0	NULL
00:00:00.00946	87	UPDATE dbo.ORDERS SET O.CARRIER_ID = @d	sa	(15ms)WRTLELOG	2	0	0	NULL
00:00:00.00890	105	DELETE dbo.NEW_ORDER WHERE NEW_ORDER	sa	(1ms)WRTLELOG	0	0	0	NULL
00:00:00.00865	65	UPDATE dbo.CUSTOMER SET C.BALANCE = @US	sa	(8ms)WRTLELOG	3	0	0	NULL
00:00:00.00860	96	UPDATE dbo.CUSTOMER SET C.BALANCE = @US	sa	(13ms)WRTLELOG	0	0	0	NULL
00:00:00.00843	88	UPDATE dbo.CUSTOMER SET C.BALANCE = @US	sa	(3ms)WRTLELOG	3	0	0	NULL
00:00:00.00840	78	UPDATE dbo.CUSTOMER SET C.BALANCE = @US	sa	NULL	3	0	0	NULL
00:00:00.00823	101	UPDATE dbo.ORDERS SET O.CARRIER_ID = @d	sa	(14ms)WRTLELOG	2	0	0	NULL
00:00:00.00763	117	UPDATE dbo.ORDERS SET O.CARRIER_ID = @d	sa	(14ms)WRTLELOG	3	0	0	NULL
00:00:00.00763	110	UPDATE dbo.ORDERS SET O.CARRIER_ID = @d	sa	(11ms)WRTLELOG	0	0	0	NULL
00:00:00.00700	104	DELETE dbo.NEW_ORDER WHERE NEW_ORDER	sa	(22ms)WRTLELOG	0	0	0	NULL
00:00:00.00700	90	DELETE dbo.NEW_ORDER WHERE NEW_ORDER	sa	(1ms)WRTLELOG	4	0	0	NULL
00:00:00.00620	64	UPDATE dbo.ORDERS SET O.CARRIER_ID = @d	sa	(7ms)WRTLELOG	0	0	0	NULL
00:00:00.00596	92	UPDATE dbo.ORDERS SET O.CARRIER_ID = @d	sa	(8ms)WRTLELOG	3	0	0	NULL
00:00:00.00483	113	UPDATE dbo.ORDERS SET O.CARRIER_ID = @d	sa	(5ms)WRTLELOG	0	0	0	NULL
00:00:00.00470	71	UPDATE dbo.ORDERS SET O.CARRIER_ID = @d	sa	(14ms)WRTLELOG	1	0	0	NULL
00:00:00.00426	91	DELETE dbo.NEW_ORDER WHERE NEW_ORDER	sa	(16ms)WRTLELOG	2	0	0	NULL
00:00:00.00423	60	DELETE dbo.NEW_ORDER WHERE NEW_ORDER	sa	(13ms)WRTLELOG	0	0	0	NULL
00:00:00.00380	75	UPDATE dbo.CUSTOMER SET C.BALANCE = @US	sa	(18ms)WRTLELOG	2	0	0	NULL
00:00:00.00346	114	UPDATE dbo.ORDERS SET O.CARRIER_ID = @d	sa	(18ms)WRTLELOG	0	0	0	NULL

sp_WhoIsActive has a lot of options

My favorites are:

- @get_task_info = 2
Displays additional wait information.
- @get_locks = 1
Collects locking information. Really helpful troubleshooting blocking storms.
- @get_avg_time = 1
Displays average times based on previous runs of the same query.
- @delta_interval = 5
Record metrics at a regular interval.

SQLintersection

Orlando, Florida – April 2014

www.SQLintersection.com

Want to find out more?

- Check out “What’s Going On in My SQL Server?”:
- <http://www.brentozar.com/archive/2011/12/whats-going-on-my-sql-server-video/>
- Kendra covers dumping sp_WhoIsActive results to a table for later analysis.



137

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

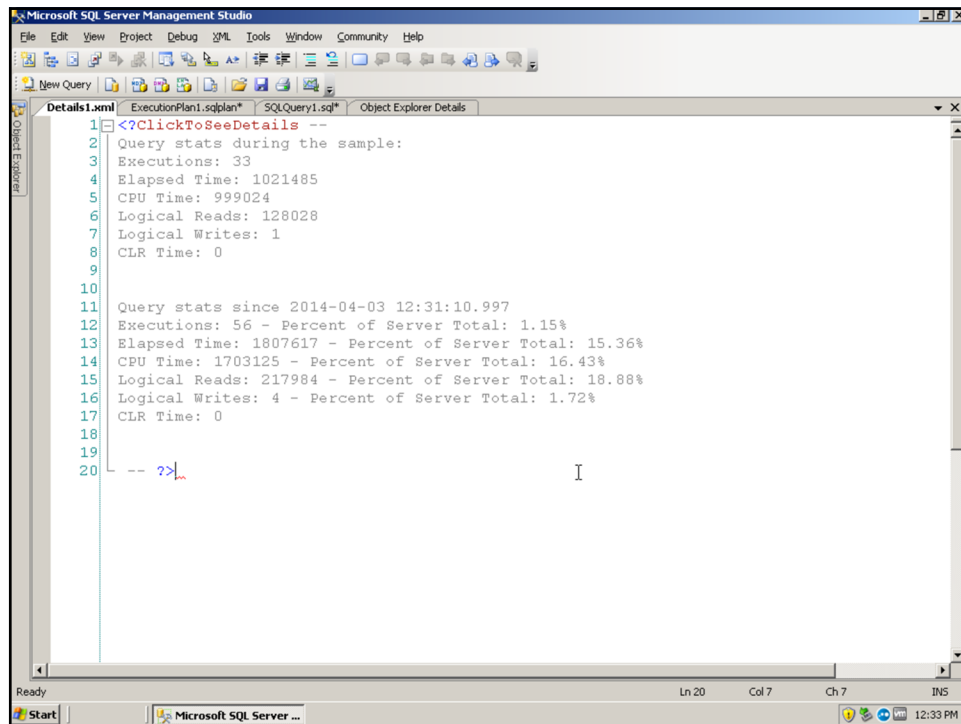
sp_AskBrent®



www.SQLIntersection.com

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Page 70



sp_AskBrent® Parameters

- **@ExpertMode = 1**
Returns all sampled metrics, not just the problems
- **@OutputDatabaseName = 'mydb',**
@OutputSchemaName = 'dbo',
@OutputTableName = 'AskBrentResults'
Writes the results to table, but clean it out frequently
- **@AsOf = '20140214 14:00',**
@OutputDatabaseName = 'mydb',
@OutputSchemaName = 'dbo',
@OutputTableName = 'AskBrentResults'
Shows the results from the log table around that time

These tools analyze
right now.
What if we can't catch
the problem?

sp_BlitzCache™



sp_BlitzCache™

- **Analyzes historical SQL Server performance:**
 - Finds problem trends.
 - Finds queries with specific problems.
 - Displays a prioritized list of things to investigate.
- **Show worst queries, sorted by reads, writes, CPU, etc.**



145

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Why analyze the plan cache?

- **Some queries may be too fast to catch any other way.**
 - A query that takes 40 microseconds is hard to catch no matter what.
 - A query with large run time variance is also difficult to spot.
- **Other query problems can be difficult to find without directly investigating plans.**
 - Working by hand is too slow.
 - Software solves the problem nicely.



146

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

DEMO!

There's one problem...

- **The plan cache holds data since (pick multiple):**
 - Last SQL Server restart
 - The last RECONFIGURE WITH OVERRIDE
 - The last recompile of a plan
 - Stats rebuild
(which should recompile all plans for a single object)
 - Memory pressure forced plans to expire
 - The last solar flare
- **In other words:**
We may not have a record of what happened.

Extended Events



Extended Events

- **Introduced with SQL Server 2008**
- **Allows lightweight monitoring of SQL Server**
 - This can start out lighter weight than Profiler
 - Allows many additional events to be added in.
- **Far more customizable than Profiler**
- **More important:**
 - Sophisticated filtering
 - Easy to collect and analyze multiple events as one workload.



The downside

- **There's no built-in GUI prior to SQL Server 2012.**
 - A plugin exists for SQL Server 2008 & R2.
 - Otherwise, you're writing the collection on your own.
- **It's not that hard, seriously.**



151

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

What We'll Set Up

- **An Extended Events trace.**
- **Capture the following events:**
 - `sp_statement_completed`
 - `sql_statement_completed`
 - `wait_info`
- **And some extra info:**
 - Client app name
 - Client host name
 - Database name
 - Plan and query information



152

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

DEMO!

[http://www.brentozar.com/go/
collect-xevents/](http://www.brentozar.com/go/collect-xevents/)

Just in case...

- **You can do something similar with ClearTrace.**
- **ClearTrace works with server side traces**
 - Set up a specific trace.
 - Process the trace when you're done.
 - Review results in the UI



155

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Why use one over the other?

- **Profiler**
 - Works anywhere
 - Allows limited data collection
 - Deprecated in 2012
- **Extended Events**
 - Can correlate many server side events
 - Allows sampling
 - Works on 2008 and newer



156

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

What we covered



Four Ways to Find Problem Queries

- **Right now:**
 - `sp_WholsActive`
 - `sp_AskBrent®`
- **Over all time:**
 - `sp_BlitzCache™`
- **On a specific workload:**
 - SQL Server Extended Events
- **Collect them all at** <http://brentozar.com/responder>



SQLIntersection
Session POSTCON01

Fix Your Worst Index Problems

Kendra Little
Kendra@BrentOzar.com



Meet Stanley

Stanley is a Senior Software Developer at Adventure Works, a bicycle dealer.



160

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Stanley hasn't been tuning
indexes regularly



161

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>



162

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Not so much

- Stanley probably has some index problems
- But how can he find them?
- We'll show him!
- You'll learn:
 - Free tools to make things easier
 - Top 3 index problems to fix first
 - Gotchas to look out for



163

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

A refresher from what you learned in “How to Think Like the Engine”

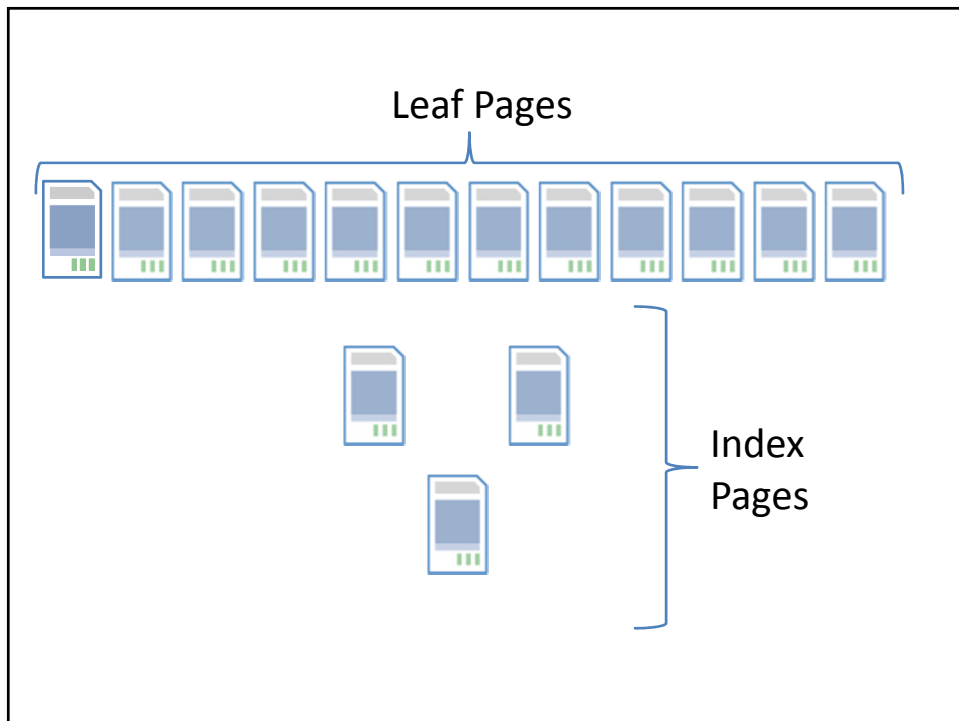
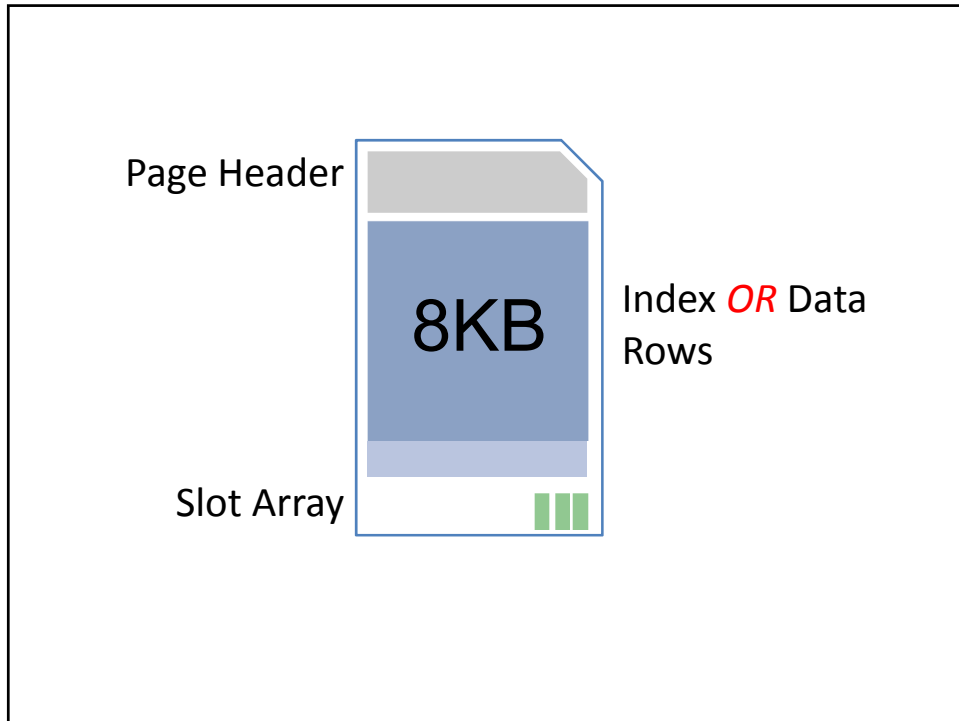
Quick recap



SQLIntersection

Orlando, Florida – April 2014

www.SQLIntersection.com



Basic indexes

▪ Clustered

- One per table
- This is the data itself (this IS the table)
 - Is ordered by the defined key
 - Also contains **all columns in the table**

▪ NonClustered

- Many per table
- “Helper” copies of PART of the table
- Could be a “covering” index or could be used for lookups



167

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Meet the DMVs

I don't drive, I just
bicycle.



Index DMV reference

- **“What are the index properties?” (Name, ID, Fillfactor, Hypothetical, disabled, filters)**
 - sys.indexes
- **“What’s the index definition?” (key and included columns)**
 - sys.index_columns
 - sys.columns
- **“How often is the index used, and in what ways?”**
 - sys.dm_db_index_usage_stats
 - sys.dm_db_index_operational_stats
- **“How big is the index?” (Rows, size, compression)**
 - sys.partitions
 - sys.dm_db_partition_stats



169

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

“Usage Stats”

- **sys.dm_db_index_usage_stats**
 - **Shows the number of executions where a plan included an operator**
 - Does NOT show if the operator was used (or how often it was accessed)
 - **Number and last date of reads**
 - Seeks
 - Scans
 - Lookups
 - **Number and last date of last write**
 - Insert/update/deletes all called “updates”
 - **Data is since startup OR since the last time the index was modified**



170

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

“Operational Stats”

- `sys.dm_db_index_operational_stats`
- Lower level, more transitory
- Lock waits
 - Page and row
- Access counts
 - Doesn't distinguish between full scans/range scans, or even range scans and seeks
- Data is only persisted while an object's metadata is in memory
- No good way when to tell it was last cleared



171

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

“Missing index” DMVs

- `sys.dm_db_missing_index_details`
- `sys.dm_db_missing_index_columns`
- `sys.dm_db_missing_index_groups`
- `sys.dm_db_missing_index_group_stats`

They tell you:

- How many time an index would have been used
- Estimated improvement the index would give
- Estimated cost of queries that wanted it

This is incredibly useful information...

But it's just raw information



172

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>



Advice is complicated

- **Missing index DMV “requests”**
 - Recommend super wide indexes
 - Don’t de-duplicate requests
 - Don’t get thrown for all types of queries
 - Get cleared at tricky times
- **Database Tuning Advisor**
 - Needs a firm hand
 - Shouldn’t be run against production
 - Can recommend truly kooky things (including for clustered indexes)



173

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

sp_BlitzIndex® shows lots of
information in one place

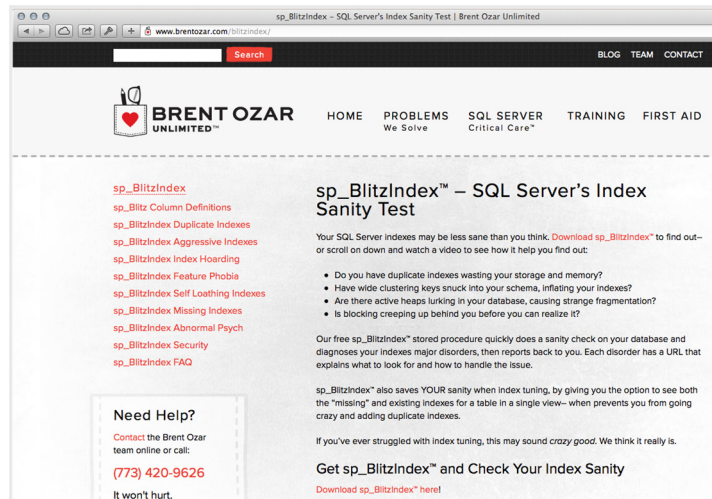


174

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>



BrentOzar.com/BlitzIndex



175

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Get details on a table

- **Examine individual tables with commands from the 'More Info' column:**

```
EXEC dbo.sp_BlitzIndex @DatabaseName='AdventureWorks',  
@SchemaName='Person',  
@TableName='Person';
```

- **This gives you:**

- Index size, use, recent locking info
- Missing index requests
- Foreign key info
- Column list (and data types)



176

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>



Demo 1: Meet the Index DMVs

Index DMVs

- **This data is useful, just don't jump to conclusions about seeks and scans**
- **Remember:**
 - Scans aren't always bad
 - Index usage stats stays around longer– but is less granular/precise
 - Operational stats aren't perfect either. And ...
- **Look at the big picture!**

Let's find out what's wrong.

How we diagnose

- **sp_BlitzIndex** (reads those DMVs!)
<http://www.brentozar.com/blitzindex>
- **Syntax to run in “diagnose” mode**
`EXEC dbo.sp_BlitzIndex @DatabaseName='AdventureWorks';`

DEMO 2: Diagnose!

Use the worksheet
on the next slide as
we go



Category	Basic description	Your notes as we work through the demo
Duplicates	Waste memory, extra IO, extra storage	
Fill Factor	Waste memory, extra IO, extra storage	
Unread Indexes	Slower writes, storage	
Active Heap	Extra random IO, weird bloat, not seek-able	
Missing Indexes	Extra IO	
Workaholics	Pointer to where the action is	

Can indexes make things faster?

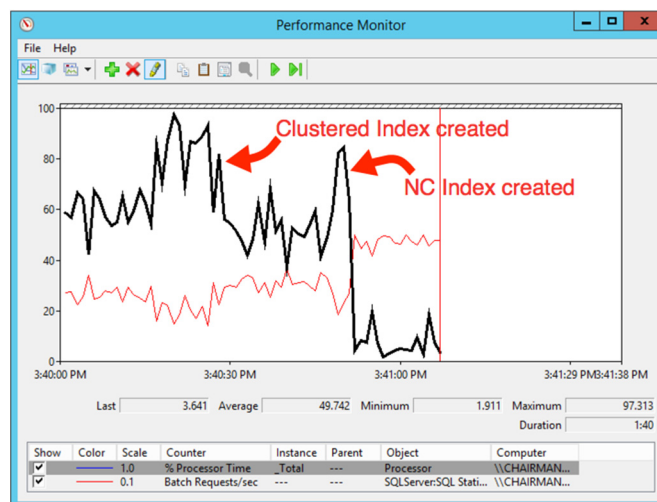
Workload	Top 3 Waits (30 second AskBrent® sample in Expert Mode)		Trans Per Minute
Original Workload	CXPACKET SOS_SCHEDULER_YIELD LATCH_EX	161.2 29.6 2.2	16,865
+ Unique CX on TransactionHistory	CXPACKET SOS_SCHEDULER_YIELD LATCH_EX	143.9 24.1 2.8	17,790
+ Covering NC on TransactionHistory	SOS_SCHEDULER_YIELD WRITELOG PAGELATCH_EX	3.3 0.7 0.0	29,900



183

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Another view



Indexing Gotchas



Missing index requests ain't always there

- **Not every query generates a missing index request**
 - "Trivial" execution plans don't generate requests
 - Some "FULL" optimizations don't either
- **If they are run very frequently, an index may be worthwhile**
 - You won't see this in missing index recommendations!
- **To find these:**
 - Use the execution plan cache
 - Not perfect– memory pressure and RECOMPILE hints impact what you will see
 - Run traces for frequently run queries

Missing Index Data Disappears Sometimes

- You can't clear this manually (no command for it)
- Like usage data, this clears at SQL Server restart
- Also cleared for an individual table when you make index changes, such as:
 - Creating a NC index
 - Dropping a NC index
- Always capture usage and missing index info BEFORE each change!



187

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Indexing Gotchas in SQL Server 2012+

- Index usage is really important, but make sure you understand when it was last cleared
 - In SQL Server 2012, this is currently cleared by any ALTER INDEX REBUILD commands
 - This was not fixed in SP1
 - There's no good way to tell when the usage was last cleared from DMVs, you can just see
 - When any metadata on the object was changed (sys.objects)
 - When SQL Server was restarted



188

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

2012+ gotchas continued

- **Readable secondaries in Availability Groups make indexes tricky to manage**
 - Reads are tracked on each instance
 - Not aggregated centrally
- **Example problem: Is that index *really* unused?**



189

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Reducing change impact

- **If you have Enterprise Edition**
 - Use ONLINE keyword when creating nonclustered indexes (unless you want to cause blocking)
 - Dropping indexes still involves blocking locks
- **Also plan how you will handle the rollback if needed**
 - Talk about this as part of the change process

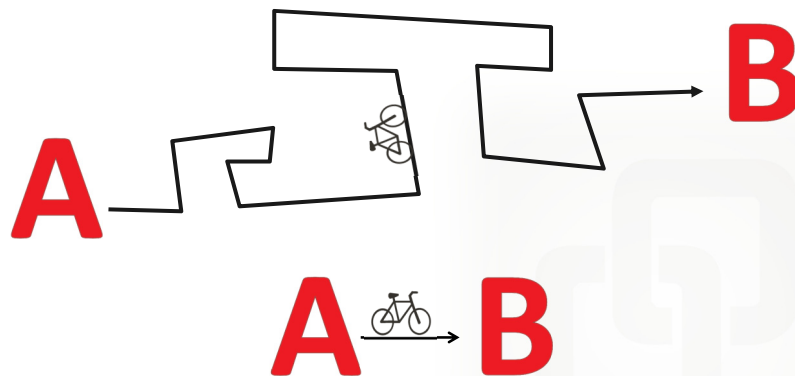


190

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Fix the top 3 problems first

Missing indexes > 1 million



Duplicate indexes > 500MB with reads

I shouldn't keep a spare index handy?



Heaps > 500MB that are being written to



Top 3

1. Missing indexes > 1 million (“magic number”)
2. Large duplicate indexes (“multiple personality”)
3. Large, active heaps



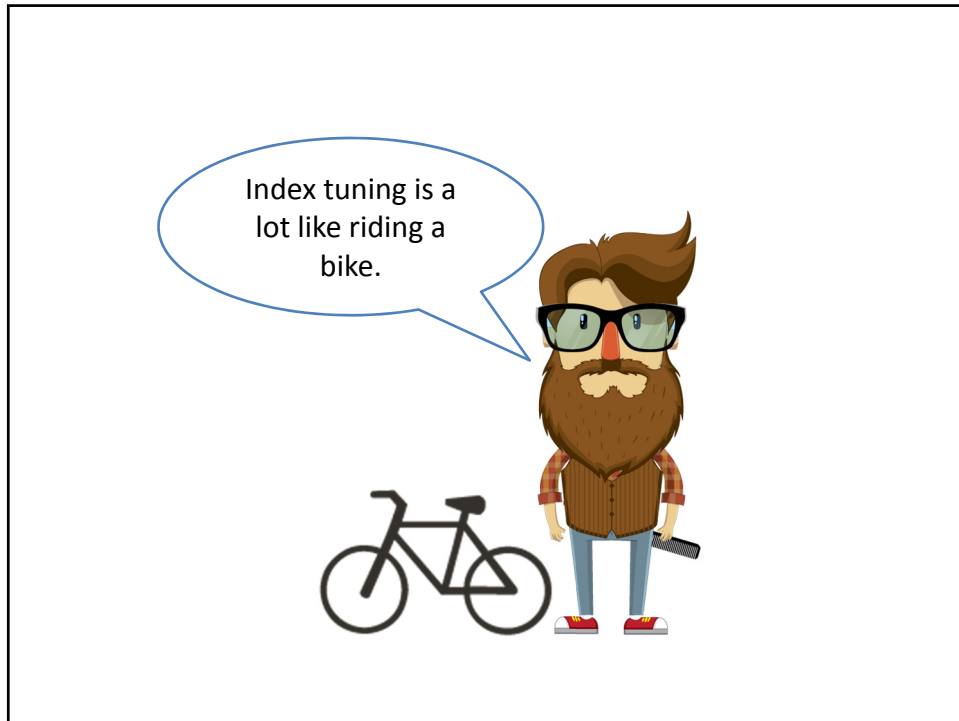
Keep an eye out for weird patterns

1. Super low fillfactor (“self loathing”)
2. Wide clustering keys with lots of nonclustered indexes (“self loathing”)
3. Identity keys that may run out of integers (“Abnormal Psychology”)
 - These are also performance killers.

SQLIntersection

Orlando, Florida – April 2014

www.SQLIntersection.com



SQLIntersection
Session POSTCON01

Artisanal Indexes: Filters, Views, and
Computed Columns

Kendra Little
Kendra@BrentOzar.com



SQLintersection

Orlando, Florida – April 2014

www.SQLintersection.com

I'm from Portland, OR

- We love things that are handcrafted
- One-of-a-kind
- Made by a highly trained artisan



199

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Indexes can benefit from hand-crafting, too

Here's the top three scenarios:

- Application frequently queries a small subset of values in a column in a large table
- Application frequently queries aggregations
- Application frequently queries computations (and you can't make the application just write the computed value in the first place)



200

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>



SQLIntersection

Orlando, Florida – April 2014

www.SQLIntersection.com

A big gotcha

- All three of these “special indexes” features share a requirement
- They won’t work with clients who are using different “SET” options in their connect string
 - Can’t use it for reads
 - Writes will fail
- This can break your application

Msg 1934, Level 16, State 1, Line 1
INSERT failed because the following SET options have incorrect settings:



201

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Settings that impact filtered indexes

SET options	Required value
ANSI_NULLS	ON
ANSI_PADDING	ON
ANSI_WARNINGS	ON
ARITHABORT	ON
CONCAT_NULL_YIELDS_NULL	ON
NUMERIC_ROUNDABORT	OFF
QUOTED_IDENTIFIER	ON



202

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>



Don't let this stop you

- **Just test before you implement these indexes**
 - Validate application activity
- **Hey, that's not so bad, is it?**
- **Just make sure you include validating that WRITES can occur as part of the testing**



203

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

```
create index ix_macaron_pink  
on dbo.macaron (color)  
where color = 'pink'
```

1. Filtered indexes



SQLintersection

Orlando, Florida – April 2014

www.SQLintersection.com

Nonclustered index

```
CREATE NONCLUSTERED INDEX  
    [TransactionHistory_ActualCost]  
    ON [Production].[TransactionHistory]  
    ([ActualCost],[TransactionID]);  
  
GO
```



205

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Filtered index

```
CREATE NONCLUSTERED INDEX  
    [TransactionHistory_ActualCost_Filtered]  
    ON [Production].[TransactionHistory]  
    ([ActualCost],[TransactionID])  
WHERE [ActualCost] >= 10;  
  
GO
```



206

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>



Filtered index with includes

```
CREATE NONCLUSTERED INDEX  
[TransactionHistory_ActualCost_Includes_Filtered]  
ON [Production].[TransactionHistory]  
([ActualCost],[TransactionID])  
INCLUDE ([ProductID],  
[ReferenceOrderID],  
[ReferenceOrderLineID],  
[TransactionDate],[TransactionType],  
[Quantity],[ModifiedDate])  
WHERE [ActualCost] >= 10;  
GO
```

Demo: Filtered Indexes

Filtered limitations

- **SQL Server 2008+**
- **Filter can only contain string literals:**
 - No functions
 - Frustrating with “ascending date” type columns
- **Statistics automatically update at the same rate as the base table (~ 20% of rows updated)**
- **Tricky to get SQL Server to use them**
 - Parameterized queries won’t consider them “safe” without trickery (stored procedures, forced parameterization)



209

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Filtered index perks

- **Works in Standard Edition**
- **Ideal case: very selective column**
 - Example: IsActive = 1 in a table with a lot of inactive rows
 - This index will be much smaller than it would be otherwise
 - Fast to read
 - Takes up less space in memory



210

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

(WHERE Bugs=1);

DBCC PAGE incorrect output with filtered indexes by Luciano Moreira

Status: **Closed**

"At the moment, there are no plans to enhance this command as it is undocumented."

BrentOzar.com/go/DBCCPageBug

Paul White: Page Free Space

A technical SQL Server blog from New Zealand.

MERGE Bug with Filtered Indexes

A MERGE statement can fail, and incorrectly report a unique key violation when:

- The target table uses a unique filtered index; and
- No key column of the filtered index is updated; and
- A column from the filtering condition is updated; and
- Transient key violations are possible

BrentOzar.com/go/FilterMergeBug



211

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

2. Indexed Views



SQLIntersection

Orlando, Florida – April 2014

www.SQLIntersection.com

Create the view

```
CREATE VIEW dbo.IndexedViewTest
WITH SCHEMABINDING
AS
SELECT
    YEAR(TransactionDate) as TransactionYear,
    MONTH(TransactionDate) as TransactionMonth,
    pr.Name as ProductName,
    SUM(Quantity) as TotalQuantity,
    SUM(ActualCost) as TotalActualCost,
    COUNT_BIG(*) as RequiredColumn
FROM Production.TransactionHistory AS th
JOIN Production.Product AS pr on
    th.ProductID=pr.ProductID
GROUP BY YEAR(TransactionDate),
    MONTH(TransactionDate),
    pr.Name
GO
```



213

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Create an index

```
CREATE UNIQUE CLUSTERED INDEX
    cx_IndexedViewTest ON
    IndexedViewTest (TransactionYear,
    TransactionMonth, ProductName);
GO
```



214

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>



Demo: Indexed Views

Indexed view fine print

- **View must be created with schemabinding**
 - References only tables in the same database
 - No outer joins, self joins, or subqueries
 - No OVER clause (or ranking/windowing functions)
 - No OUTER or CROSS APPLY
- **If you're grouping (most common use)**
 - The view must contain a COUNT_BIG column
 - No HAVING
 - No MIN, MAX, TOP, or ORDER BY
- **The clustered index on the view must be unique**
- **Lots more rules – Books Online has a full list:**
BrentOzar.com/go/viewrules

Indexed view limitations

- “Why isn’t SQL using my indexed view?” ← asks everyone who tries this feature
- There’s an Enterprise Edition feature to match queries with the view automatically
 - Watch your query plans closely– it may not use the view as often as you want
 - Even in EE, you may often need to reference the view by name and add a NOEXPAND hint
- Every indexed view (and nonclustered index you add to an indexed view) adds overhead:
 - More writes for each insert/update/delete
 - More to check for corruption



217

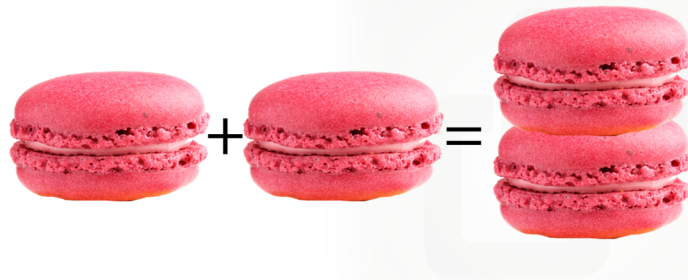
© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Indexed view perks

- Works in Standard Edition (with NOEXPAND hint)
- Works in SQL Server 2005+
- Dramatically reduces IO when used correctly



3. Computed columns



Example

```
CREATE TABLE [Sales].[SalesOrderHeader]
([SalesOrderID] [int] IDENTITY(1,1),
 [SubTotal] [money],
 [TaxAmt] [money],
 [Freight] [money],
 [TotalDue] AS
    (isnull
      (([SubTotal]+[TaxAmt])+[Freight]
      ,(0)))
)
GO
```

SQLIntersection

Orlando, Florida – April 2014

www.SQLIntersection.com

You can “persist”

```
CREATE TABLE [Sales].[SalesOrderHeader]
([SalesOrderID] [int] IDENTITY(1,1),
 [SubTotal] [money],
 [TaxAmt] [money],
 [Freight] [money],
 [TotalDue] AS
    (isnull
        (([SubTotal]+[TaxAmt])+[Freight]
        ,(0))) PERSISTED
)
GO
```



221

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

You can index

```
CREATE TABLE [Sales].[SalesOrderHeader]
([SalesOrderID] [int] IDENTITY(1,1),
 [SubTotal] [money],
 [TaxAmt] [money],
 [Freight] [money],
 [TotalDue] AS
    (isnull
        (([SubTotal]+[TaxAmt])+[Freight]
        ,(0)))
)
GO

CREATE INDEX k1_SalesOrderHeader_TotalDue
on [Sales].[SalesOrderHeader](TotalDue)
INCLUDE (SalesOrderId);
GO
```

Note: This also makes the column PERSISTED



222

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>



Demo:

Indexed Computed Columns

Here's the magic

- **Queries can pick up the indexed computed column without directly referring to it by name**
 - SQL Server matches the computation in the query to the computation in the computed column
 - Then it can use the index on it
- **Computed columns can also generate statistics that improve estimates and get better plans**
- **This allow you to optimize some queries without changing their syntax!**

SQLIntersection

Orlando, Florida – April 2014

www.SQLIntersection.com

Clustered index:

[PK_SalesOrderHeader_SalesOrderID]

Key Column(s)	Other Columns (some removed for display purposes)...								
SalesOrderID	Revision Number	Order Date	Due Date	SubTotal	TaxAmt	Freight	TotalDue = isnull(((SubTotal)+[TaxAmt])+[Freight],(0))	Comment	ModifiedDate
43659	3	2005-07-01	2005-07-13	20565.6206	1971.5149	616.0984	23153.2339	NULL	2005-07-08 00:00:00.000
43660	3	2005-07-01	2005-07-13	1294.2529	124.2483	38.8276	1457.3288	NULL	2005-07-08 00:00:00.000
43661	3	2005-07-01	2005-07-13	32726.4786	3153.7696	985.553	36865.8012	NULL	2005-07-08 00:00:00.000

NonClustered Index on computed column:

SalesOrderHeader_TotalDue

Key Column(s)	
TotalDue = isnull(((SubTotal)+[TaxAmt])+[Freight],(0))	SalesOrderID
1457.3288	43660
23153.2339	43659
36865.8012	43661

← "Secret" column

↑ Index statistics created on persisted values

This query is covered

```
SELECT
    SalesOrderID
FROM Sales.SalesOrderHeader
WHERE (isnull(((SubTotal)+[TaxAmt])+[Freight],(0))) > 100;
```

Key Column(s)	
TotalDue = isnull(((SubTotal)+[TaxAmt])+[Freight],(0))	SalesOrderID
1457.3288	43660
23153.2339	43659
36865.8012	43661



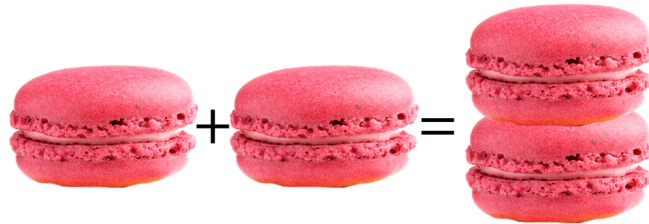
226

© SQLIntersection. All rights reserved.
http://www.SQLIntersection.com



Indexed computed column perks

- Works in Standard Edition
- Works in SQL Server 2005+
- Can optimize code without requiring code changes in the client application



227

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Summing up

Artisanal Indexes



Hand crafted indexes

Here's how to recognize when to use these:

1. Application frequently queries a small subset of values in a column in a large table: Filtered indexes
2. Application frequently queries an aggregation on low or moderate write tables: Indexed views
3. Application frequently queries computations (and you can't make the application just write the computed value in the first place): Indexed computed columns



229

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Remember to test first

- These features persist filtered/computed/aggregated values
- SQL Server needs to make sure that the values are computed in a way that matches the client
- That's why 'SET' options are so important!

```
Msg 1934, Level 16, State 1, Line 1
INSERT failed because the following SET options have incorrect settings:
```



230

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Don't overuse special indexes

- 99% of the time, you need plain old Clustered and NonClustered indexes
- Don't overload on indexes – adding indexed views (and then indexes on the indexed view) piles up fast
- Don't force yourself into using 'recompile' for lots of your queries to get a filtered index to be used
- Don't use indexed computed columns if you could just make the app compute and write the value into a column itself



231

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

When you find the right place to implement a perfectly hand crafted index, *it's delicious.*



232

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

SQLIntersection

Orlando, Florida – April 2014

www.SQLIntersection.com

SQLIntersection Session POSTCON01

Developer's Guide to Dangerous Queries

Jeremiah Peschka
jeremiah@BrentOzar.com



What bad patterns have we covered?

- Using built-in functions in the WHERE clause
- Full wildcards with LIKE
- Implicit Conversions
- Locking and blocking
- Residual predicates
- Plan compilation time out



234

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>



We've already shown you why

- **Indexes matter**
- **Everything can ruin SARGability**
 - Functions can mask data
 - LIKE statements can require scans
 - Implicit conversions are bad
- **Locking and blocking can ruin your day**



235

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Where to now?



Our Agenda

- Low Functioning Functions
- Nesting Habits of the Wild Query
- Blueprint Blunders



237

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Low Functioning Functions



Scalar Functions

A function
that returns
a single value

SQLIntersection

Orlando, Florida – April 2014

www.SQLIntersection.com

What could go wrong?

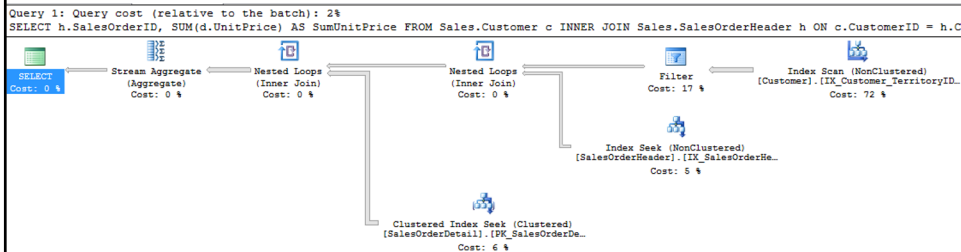
```
SELECT h.SalesOrderID,
       SUM(d.UnitPrice) AS SumUnitPrice
FROM   Sales.Customer c
       INNER JOIN Sales.SalesOrderHeader h
           ON c.CustomerID = h.CustomerID
       INNER JOIN Sales.SalesOrderDetail d
           ON h.SalesOrderID = d.SalesOrderID
WHERE  dbo.CheckPermissions(c.CustomerID) = 1
GROUP BY h.SalesOrderID;
```



241

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

What could go wrong?



242

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>



What could go wrong?

- Table 'SalesOrderDetail'.
Scan count 861, logical reads 2765
- Table 'SalesOrderHeader'.
Scan count 121, logical reads 243
- Table 'Customer'.
Scan count 1, logical reads 37



243

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

What could go wrong?

- Table 'SalesOrderDetail'.
Scan count 861, logical reads 2765
- Table 'SalesOrderHeader'.
Scan count 121, logical reads 243
- Table 'Customer'.
Scan count 1, logical reads 37



244

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Wait, what?

- **CheckPermissions forces row by row execution.**
 - Increases access to underlying tables.
 - No easy way to tell without in-depth investigation.
- **This is going to be hard to find**
 - Average CPU per execution is 75 microseconds.
 - 6 reads per execution.



245

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Identifying the Problem: Interactive Means

- **Service Side Trace**
 - Requires the problem to be observed.
 - This issue won't show up in most traces: short duration, low reads.
- **Extended Events**
 - Finer grained control.
 - Same problems as a trace.



246

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Identifying the Problem: Historical Means

- **Plan Cache**
 - Historical information.
 - Lightweight querying.
 - We don't have to prepare in advance to capture problems.
- **I like this approach**
 - It's lightweight.
 - It lets me find problems I didn't think to watch for.



247

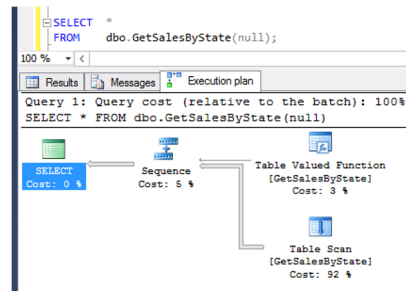
© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Demo!

What about TVFs?

A function
that returns
a tabular value

A TVF in Action!



251

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

A TVF in Action!

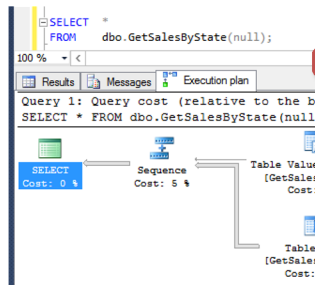


Table Scan	
Scan rows from a table.	
Physical Operation	Table Scan
Logical Operation	Table Scan
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Actual Number of Rows	6669
Actual Number of Batches	0
Estimated I/O Cost	0.003125
Estimated Operator Cost	0.003392 (92%)
Estimated CPU Cost	0.000267
Estimated Subtree Cost	0.003392
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows	100
Estimated Row Size	87 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	False
Node ID	3
Object	
[AdventureWorks2012_CS].[dbo].[GetSalesByState]	
Output List	
[AdventureWorks2012_CS].[dbo].[GetSalesByState].MonthFirstDate,	
[AdventureWorks2012_CS].[dbo].[GetSalesByState].StateProvinceName,	
[AdventureWorks2012_CS].[dbo].[GetSalesByState].TotalSales	



252

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

A TVF in Action! Now with different parameters

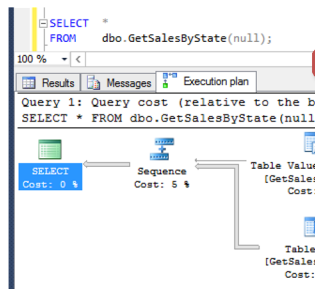


Table Scan	
Scan rows from a table.	
Physical Operation	Table Scan
Logical Operation	Table Scan
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Actual Number of Rows	37
Actual Number of Batches	0
Estimated I/O Cost	0.003125
Estimated Operator Cost	0.003392 (92%)
Estimated CPU Cost	0.000267
Estimated Subtree Cost	0.003392
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows	100
Estimated Row Size	87 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	False
Node ID	3
Object	
[AdventureWorks2012_CS].[dbo].[GetSalesByState]	
Output List	
[AdventureWorks2012_CS].[dbo].[GetSalesByState].MonthFirstDate,	
[AdventureWorks2012_CS].[dbo].[GetSalesByState].StateProvinceName,	
[AdventureWorks2012_CS].[dbo].[GetSalesByState].TotalSales	

TVFs Give Bad Estimates

- Regardless of parameter, we get the same estimate.
- If you only have a few rows, don't worry.
- If you have a lot of rows:
 - Dump rows into temporary tables
 - Re-write the TVF to inline

An Inline TVF

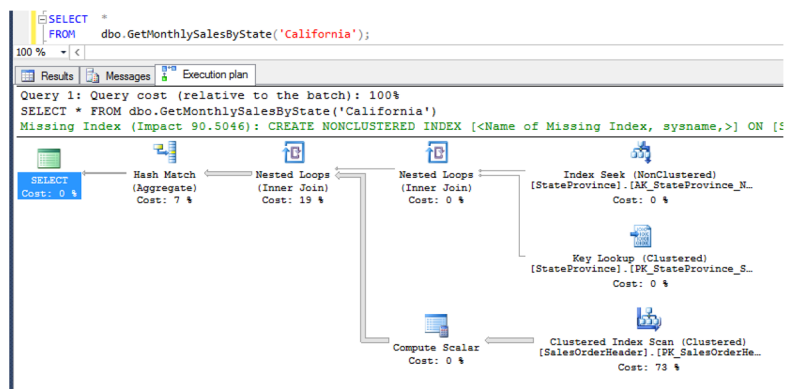
- A single statement TVF can be inlined by the optimizer.
- When inlined:
 - The body of the TVF is re-written into the query.
 - We get good estimates.
 - Things can run faster.
 - Everyone wins, yay!



255

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

An Inlined TVF

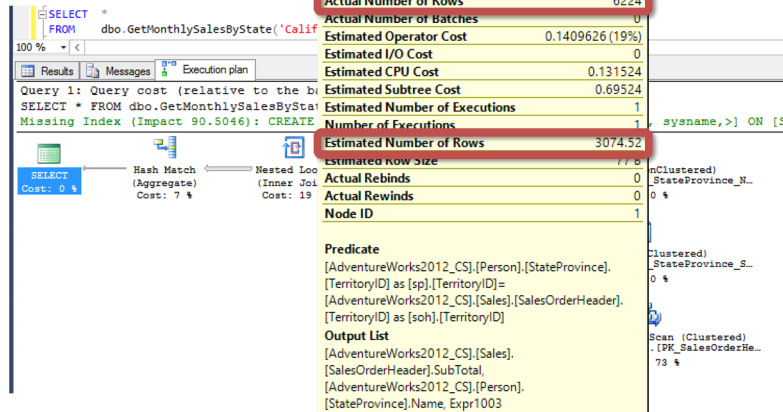


256

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>



That's better!



257

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Functions in SQL Server

- **Pros:**
 - Encapsulate logic
 - Re-usable
- **Cons:**
 - Provide inaccurate estimates
 - Hide I/O requirements
 - Obscure the cost of operations



258

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>



Nesting Habits of the Wild Query



Correlated Subqueries

What is a correlated subquery?

- A subquery that uses values from an outer query.



261

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Problems?

- **Extra reads and CPU time**
 - SQL Server may run the subquery once per row in the outer query.
- **Sudden offset**
 - Great in development, terrible in production.
 - Difficult to identify without careful observation.



262

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>



SQLIntersection

Orlando, Florida – April 2014

www.SQLIntersection.com

Our awful query

```
SELECT SalesOrderID, SubTotal,
       OrderYear, OrderMonth
FROM   ( SELECT SalesOrderID, SubTotal,
               YEAR(OrderDate) AS OrderYear,
               MONTH(OrderDate) AS OrderMonth
         FROM   Sales.SalesOrderHeader
       ) AS x
WHERE  x.SubTotal >
      ( SELECT AVG(soh.SubTotal)
        FROM   Sales.SalesOrderHeader soh
        WHERE  YEAR(soh.OrderDate) =
                x.OrderYear
              AND MONTH(soh.OrderDate) =
                x.OrderMonth
      )
```

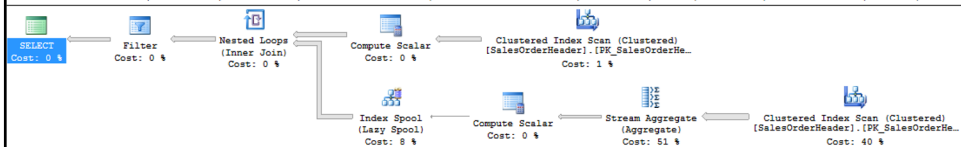


263

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

What's going on here?

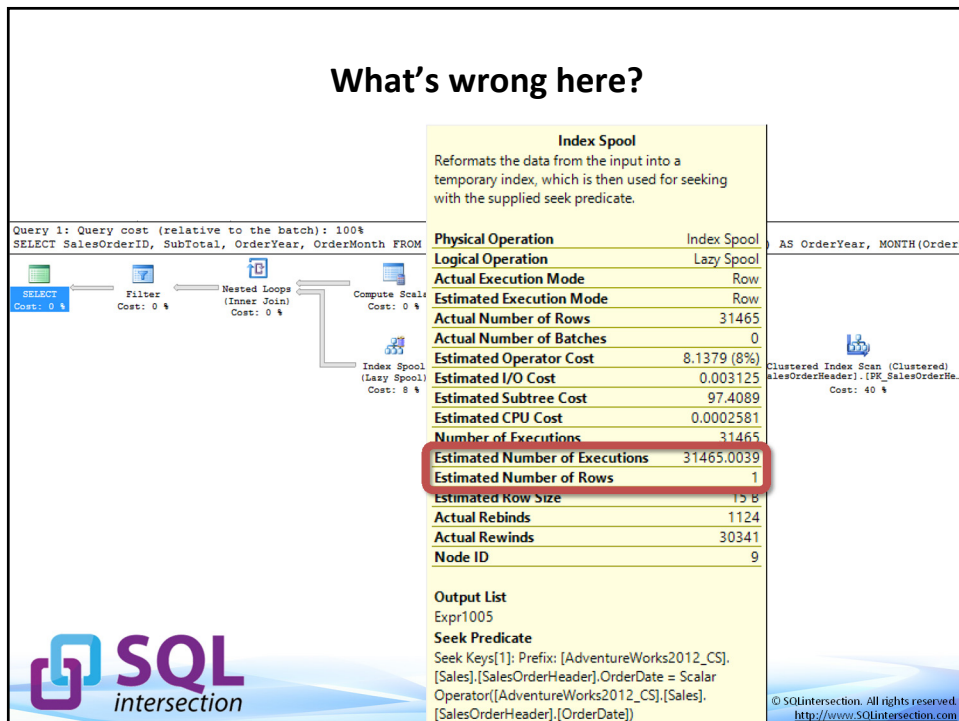
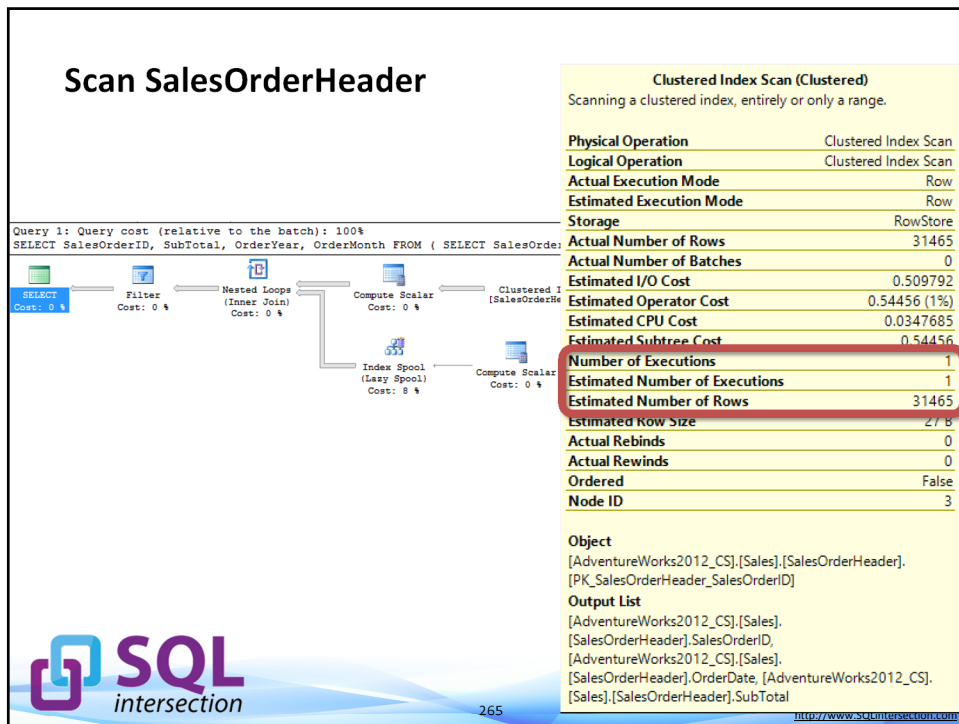
Query 1: Query cost (relative to the batch): 100%
SELECT SalesOrderID, SubTotal, OrderYear, OrderMonth FROM (SELECT SalesOrderID, SubTotal, YEAR(OrderDate) AS OrderYear, MONTH(OrderDate) AS OrderMonth FROM Sales.SalesOrderHeader) AS x WHERE x.SubTotal > (SELECT AVG(soh.SubTotal) FROM Sales.SalesOrderHeader soh WHERE YEAR(soh.OrderDate) = x.OrderYear AND MONTH(soh.OrderDate) = x.OrderMonth)



264

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>





What's wrong here?

- Table 'Worktable'.
Scan count 32589, logical reads 69792
- Table 'SalesOrderHeader'.
Scan count 1125, logical reads 775125
- We read 6,759,336KB to perform this query



267

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

A re-write!

```
SELECT SalesOrderID, SubTotal,
       OrderYear, OrderMonth
FROM   (
        SELECT SalesOrderID,
               SubTotal,
               YEAR(soh.OrderDate) OrderYear,
               MONTH(soh.OrderDate) OrderMonth,
               AVG(SubTotal) OVER
               (PARTITION BY YEAR(soh.OrderDate),
                             MONTH(soh.OrderDate)
                ) AS AverageSale
        FROM   Sales.SalesOrderHeader soh
        ) AS x
WHERE  SubTotal > AverageSale ;
```



268

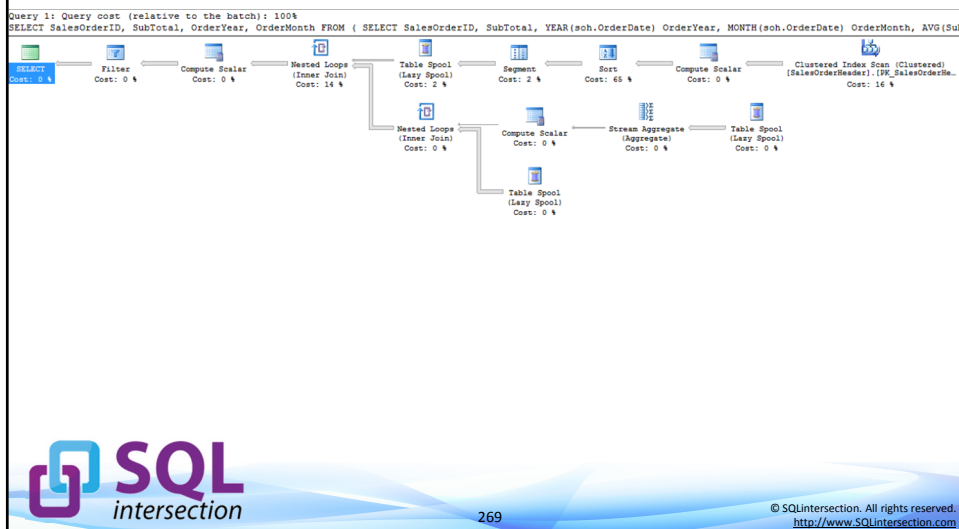
© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

SQLIntersection

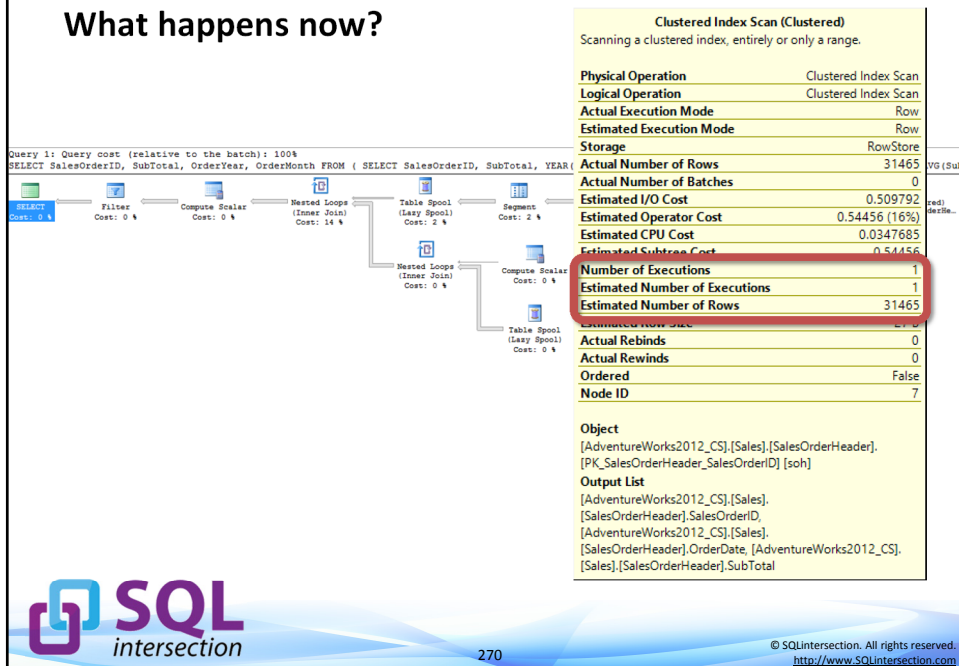
Orlando, Florida – April 2014

www.SQLIntersection.com

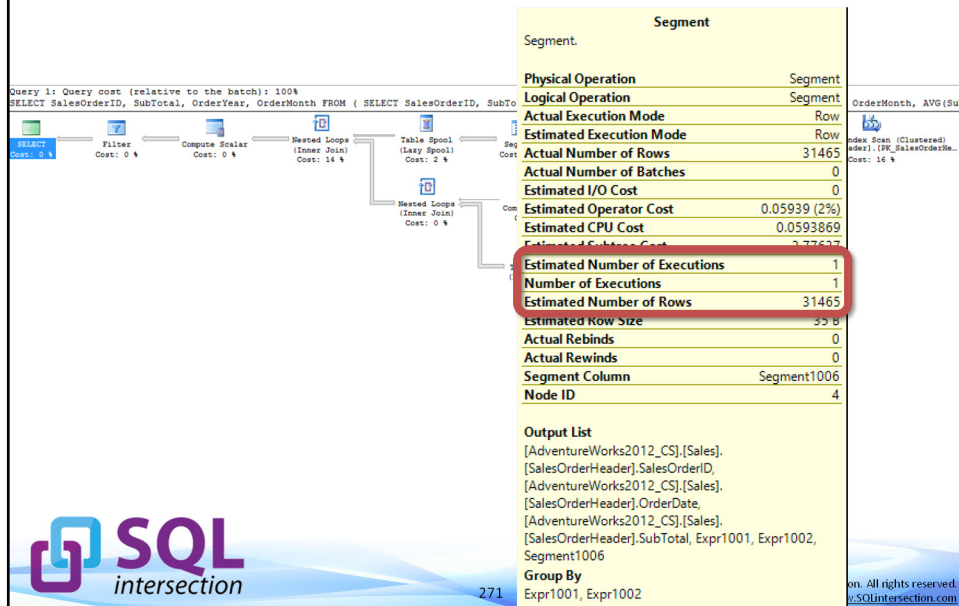
What happens now?



What happens now?



Everything runs once!



What's going on here?

- Table 'Worktable'.
Scan count 3, logical reads 64141
- Table 'SalesOrderHeader'.
Scan count 1, logical reads 689
- We've reduced the amount of I/O to 518,640KB

Simple Rewrites Save Time

- **By re-writing the query slightly differently:**
 - SQL Server produces a better plan.
 - We read the table only once.
 - The code is, arguably, more maintainable.
- **Most importantly:**
 - We eliminate multiple executions of one code fragment.



273

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Nested Views



SQLIntersection

Orlando, Florida – April 2014

www.SQLIntersection.com

It's so simple!

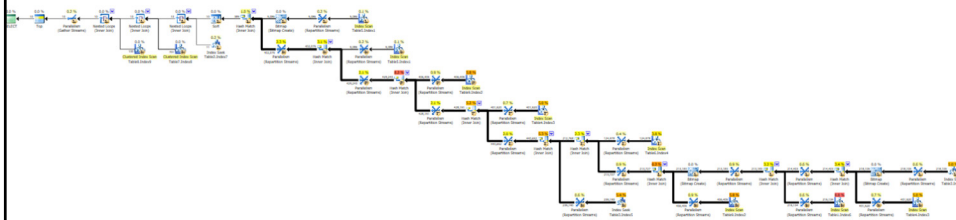
```
SELECT TOP 10 *  
FROM [dbo].[ILookReallySimple]  
WHERE Column10 = @Column10Value  
OR Column11 = @Column11Value  
ORDER BY ADateColumn DESC
```



275

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

... What?



276

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>



The problem with nested views

- It's not directly a problem with nested views.
- It's a problem of query complexity.
 - 2 tables – 2 ways to join
 - 3 tables – 9 ways to join
 - 4 tables – 64 ways to join
 - ...
 - 10 tables – 1,000,000,000 ways to join
- At some point the optimizer will give up and return the best plan it has right now.

This is just the total number of possible join combinations and doesn't include optimizer smarts



277

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Finding Exhausted Optimizers

Optimization Level	FULL
Parameter List	Column14, Column13
Physical Operation	
QueryHash	0xF71AAAB5149FC515
QueryPlanHash	0x4D9C241222B6733D
Reason For Early Termination C	Time Out



278

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>



Getting Rid of Nested Views

- **Break out the query.**
- **Re-write using temp tables.**
 - Insert into temp table from view.
 - Join between temp tables.
 - May appear more complex, almost always faster.
- **Only use the tables you need.**
 - Track down the columns you need.
 - Re-write your query to avoid views.
 - May not always be possible, depending on permissions.



279

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Why no working example?

- **We didn't produce an example in AdventureWorks that did the wrong thing.**
- **In fact, in trying to "do the wrong thing" we made our queries faster.**
- **Want to prove it out for yourself?**
 - Open up SSMS and get cracking!
 - Look in your own code base.
 - This is really time consuming if you're starting from scratch.



280

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Database Design Anti-Patterns



Why Database Design?

- **Bad design choices**
 - Last forever.
 - Cause strange querying decisions.
 - Make you cringe in 6 months.

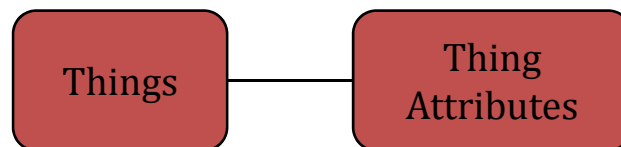


282

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Entity Attribute Value

Oh the horror!



What's in an attribute?

ThingID
AttributeName
AttributeValue

- Guess which columns are VARCHAR



285

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Problems with basic EAV

- There's minimal integrity.
- Can you ensure that
 - There are no Status attributes with a value of 'QDASLKFAJSF'?
 - Number of Children is numeric?
- Almost every query requires a PIVOT.
- Complex predicates are very difficult



286

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

A mildly better approach

ThingID
AttributeName
AttributeStringValue
AttributeDateValue
AttributeNumericValue
AttributeLobValue



287

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

A mildly better approach

ThingID
AttributeName
AttributeStringValue
AttributeDateValue
AttributeNumericValue
AttributeLobValue

At least we can index it...

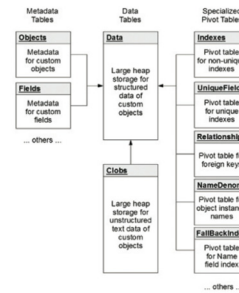


288

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

This is not an easy problem

- Salesforce went so far as to publish a whitepaper on how they make things work:
The Force.com Multitenant Architecture
- It boils down to:
 - Indexable structures
 - Rich metadata
 - Materialized views

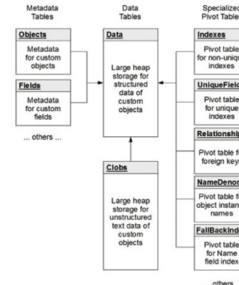


289

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

This is not an easy problem

- Salesforce went so far as to publish a whitepaper on how they make things work:
The Force.com Multitenant Architecture
- It boils down to:
 - Indexable structures
 - Rich metadata
 - Materialized views
 - Excessive query hinting



290

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Solving the problem - 1

- **Semi-structured data**
 - Materialize the most common columns
 - Store unknowns in a single column (NVARCHAR or XML)
- **This prevents many of the problems of an EAV model.**
- **This leaves you with unindexable blob data.**
- **You can mitigate the indexing with FTS or a third party FTS tool (like Lucene or SOLR).**



291

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

Solving the problem - 2

- **Post processing**
 - Stick with EAV for storing raw data.
 - Create additional processes to materialize data.
- **This can be accomplished with:**
 - Materialized views.
 - Async triggers.
 - Moving logic into the application tier.
- **This will almost certainly require scaling with hardware.**



292

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

What Can We Do?



Patterns and Anti-Patterns

- Understand the data access patterns of your code.
- Write code for convenience.
- Tune for performance.



294

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

SQLintersection
Session POSTCON01

Make SQL Server Apps Go Faster

Brent Ozar, Jeremiah Peschka, and Kendra Little
Help@BrentOzar.com



Question:
What is *the #1 best thing* you learned this week?

Your top-secret page for the scripts
we showed today (and more):

<http://BrentOzar.com/go/zoom>

Password: **notslower**

298

When you get home

- **Analyze wait stats**
- **Find the queries & tables causing the problem**
 - Sp_WhoIsActive
 - Top queries in plan cache
 - Extended Events
- **Tune, test, and deploy**
 - Weed out the common T-SQL antipatterns
 - Improve the indexes and the data design antipatterns
- **Start the process again with wait stat analysis**



298

© SQLintersection. All rights reserved.
<http://www.SQLintersection.com>

How to Get Help

- **Wait stat worries? Execution plan oddities? Post it:**
 - Developers: <http://StackOverflow.com>
 - DBAs: <http://DBA.StackExchange.com>,
or <http://SQLServerCentral.com>
- **No answers in an hour?**
 - Tweet the link with #SQLhelp.
- **No answers after a day?**
 - Email the link to Help@BrentOzar.com.
 - Mention that you were in this workshop!



299

© SQLIntersection. All rights reserved.
<http://www.SQLIntersection.com>

Please evaluate



Don't forget to complete an online evaluation on EventBoard!

Session: POSTCON01

Make SQL Server Apps Go Faster

Your evaluation helps organizers build better conferences,
and helps speakers improve their sessions.



Thank you!

