PASS
Data Community
SUMMIT
2023

Unlocking the Power of WhoIsActive

Jeff Iannucci

He/Him

Senior Consultant

Straight Path Solutions

Solving Real World SQL Server Problems

Learning Pathway

Wednesday

- Unlocking the Power of WhoIsActive – *Jeff Iannucci*
- Is Storage the Root Cause of Your Performance Woes... or Not? – *Andy Yun*

Thursday

- Mastering Dynamic SQL – *Deborah Melkin*
- Making the Most of Query Store in the Real World – *Jeff Iannucci*

Friday

- Conquering the Monster Proc – How to Combat Legacy Code – *Deborah Melkin*

The goal: make you feel like a wizard!

Who in the world is Jeff Iannucci?

StraightPath Solutions

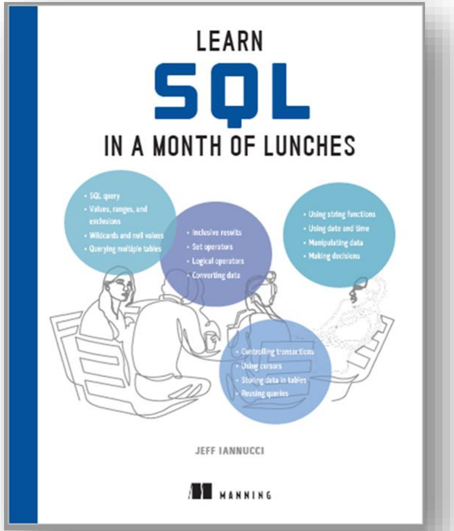
Consultant at Straight Path Solutions

PLURALSIGHT

Content Author at Pluralsight

MANNING

Author of “Learn SQL in a Month of Lunches”



LEARN SQL IN A MONTH OF LUNCHES

- SQL query
- Values, ranges, and exclusions
- Wildcards and null values
- Querying multiple tables

- Inclusive results
- Set operators
- Logical operators
- Converting data

- Using string functions
- Using date and time
- Manipulating data
- Making decisions

- Controlling transactions
- Using cursors
- Storing data in tables
- Reusing queries

JEFF IANNUCCI

MANNING

Who wrote WhoIsActive?

Adam Machanic

https://github.com/amachanic/sp_whoisactive/releases

What is WhoIsActive?

sp_WhoIsActive stored procedure

Uses over a dozen Dynamic Management Views

23 default columns returned

25 parameters that (almost) nobody uses

What versions can use WhoIsActive?

All versions from 2005 through 2019

…but check database compatibility levels

Azure SQL Database

http://dataeducation.com/sp_whoisactive-for-azure-sql-database-attempt-2/

What permissions for WhoIsActive?

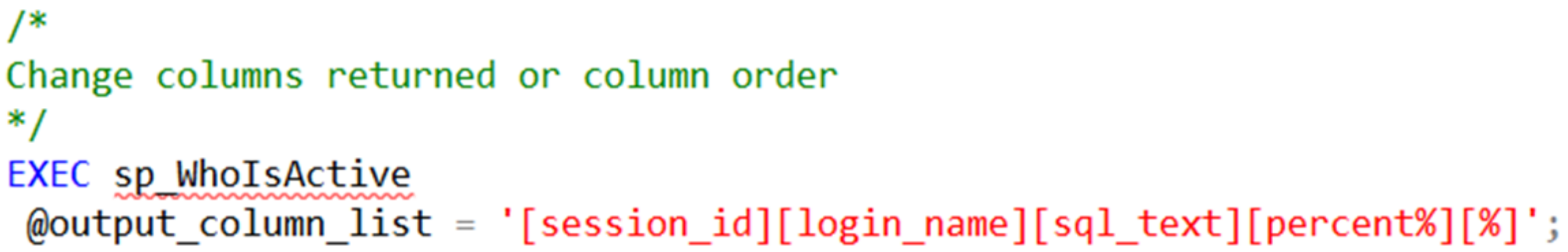
Grant VIEW SERVER STATE

Create it in the master database

Logins may require EXECUTE permissions

The basics

@output_column_list



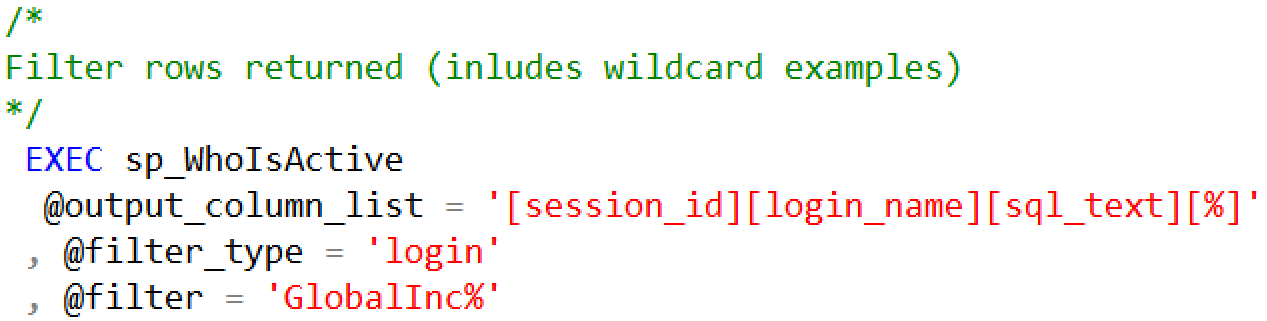
```
/*
Change columns returned or column order
*/
EXEC sp_WhoIsActive
 @output_column_list = '[session_id][login_name][sql_text][percent%][%]';
```

Use brackets, not commas

% wildcard works with column names

[%] returns all remaining columns

@filter_type & @filter



```
/*
Filter rows returned (inludes wildcard examples)
*/
 EXEC sp_WhoIsActive
  @output_column_list = '[session_id][login_name][sql_text][%]'
 , @filter_type = 'login'
 , @filter = 'GlobalInc%'
```

Filter on:

Login

Session

Host

Program

Database*

@not_filter_type & @not_filter

Exclusions!

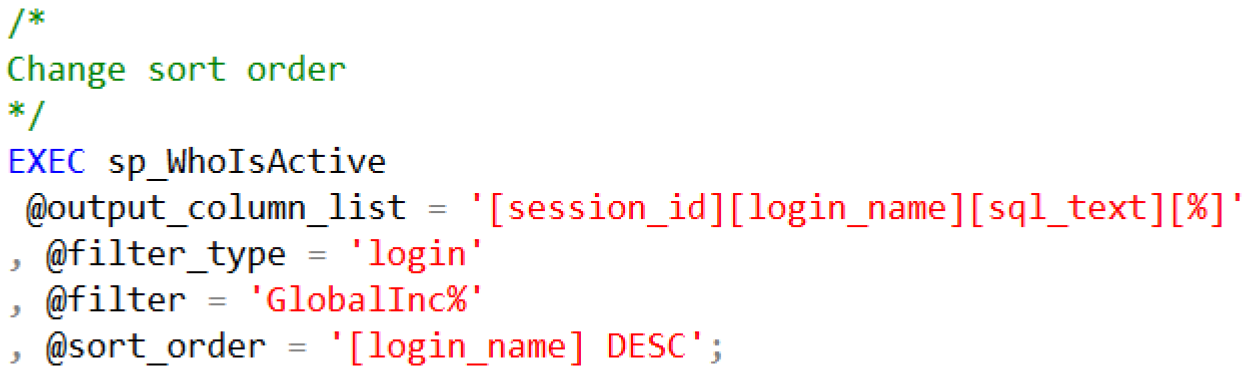
Filter on:

Login Session

Host Program

Database*

@sort_order



```
/*
Change sort order
*/
EXEC sp_WhoIsActive
 @output_column_list = '[session_id][login_name][sql_text][%]'
, @filter_type = 'login'
, @filter = 'GlobalInc%'
, @sort_order = '[login_name] DESC';
```

Use brackets, not commas

ASC or DESC order

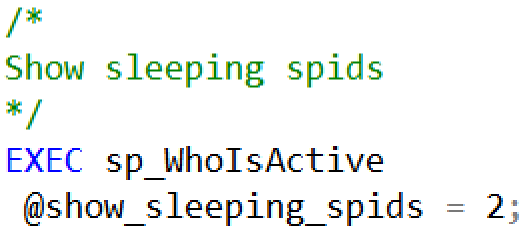
Wildcards are NOT supported

(just like SQL...)

SELECT Col1, Col2, Col3	**@output_column_list**
FROM SomeTable	
WHERE Col1 = 'X'	**@filter_type & @filter**
AND Col2 <> 'Y'	**@not_filter_type & @not_filter**
ORDER BY Col3 DESC	**@sort_order**

Hidden gems

@show_sleeping_spids



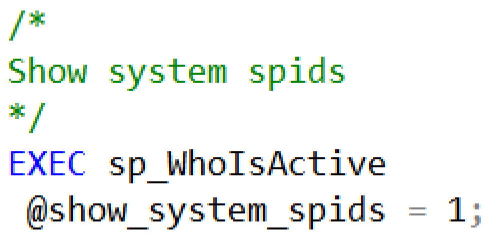
```
/*
Show sleeping spids
*/
EXEC sp_WhoIsActive
 @show_sleeping_spids = 2;
```

2 = all sleeping SPIDs

1 = only sleeping SPIDs with open transactions (default)

0 = no sleeping SPIDs

@show_system_spids



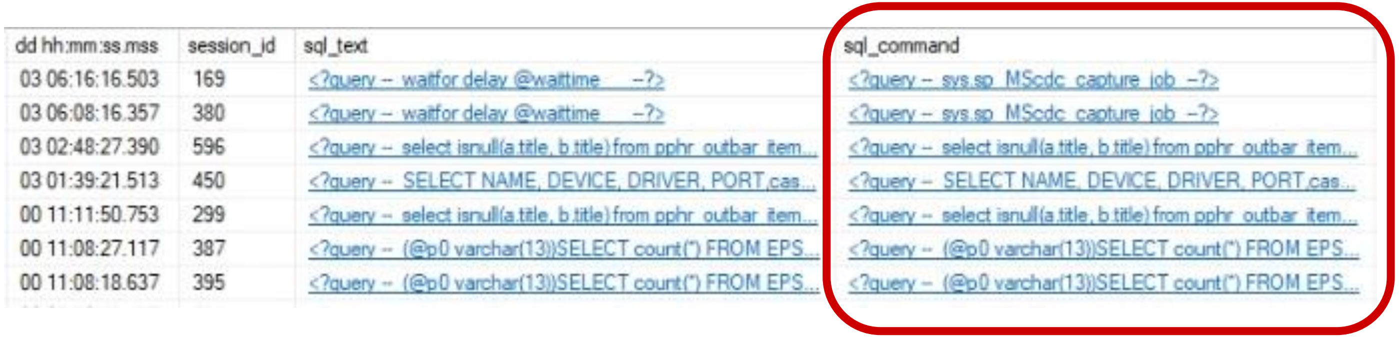
```sql
/*
Show system spids
*/
EXEC sp_WhoIsActive
 @show_system_spids = 1;
```

1 = show all system SPIDs

Use only if you’re totally confounded

@get_outer_command

```sql
/*
Show the origial command/stored procedure that called the SQL statement
*/
EXEC sp_WhoIsActive
 @get_outer_command = 1;
```

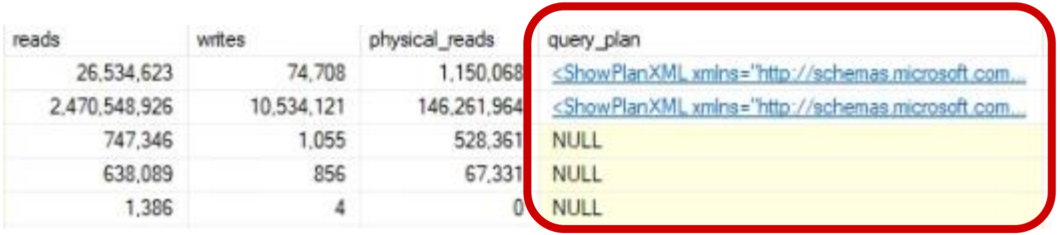


dd hh:mm:ss.mss	session_id	sql_text	sql_command
03 06:16:16.503	169	<?query -- waitfor delay @waittime --?>	<?query -- sys.sp_MScdc_capture_job --?>
03 06:08:16.357	380	<?query -- waitfor delay @waittime --?>	<?query -- sys.sp_MScdc_capture_job --?>
03 02:48:27.390	596	<?query -- select isnull(a.title, b.title) from pphr_outbar_item...	<?query -- select isnull(a.title, b.title) from pphr_outbar_item...
03 01:39:21.513	450	<?query -- SELECT NAME, DEVICE, DRIVER, PORT,cas...	<?query -- SELECT NAME, DEVICE, DRIVER, PORT,cas...
00 11:11:50.753	299	<?query -- select isnull(a.title, b.title) from pphr_outbar_item...	<?query -- select isnull(a.title, b.title) from pphr_outbar_item...
00 11:08:27.117	387	<?query -- (@p0 varchar(13))SELECT count(*) FROM EPS...	<?query -- (@p0 varchar(13))SELECT count(*) FROM EPS...
00 11:08:18.637	395	<?query -- (@p0 varchar(13))SELECT count(*) FROM EPS...	<?query -- (@p0 varchar(13))SELECT count(*) FROM EPS...

Shows outer batch/stored procedure

@get_plans

```
/*
Show the plan for the current statement
*/
EXEC sp_WhoIsActive
 @get_plans = 1;
```



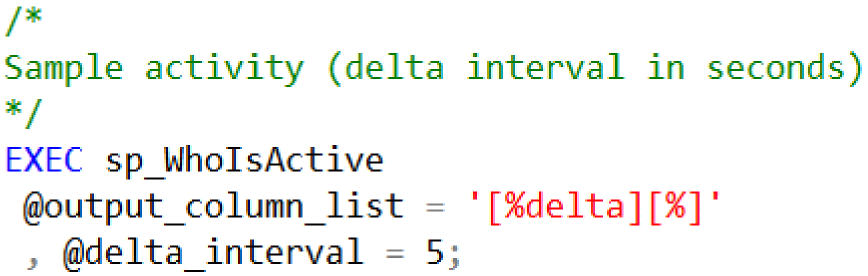
reads	writes	physical_reads	query_plan
26,534,623	74,708	1,150,068	<ShowPlanXML xmlns="http://schemas.microsoft.com...
2,470,548,926	10,534,121	146,261,964	<ShowPlanXML xmlns="http://schemas.microsoft.com...
747,346	1,055	528,361	NULL
638,089	856	67,331	NULL
1,386	4	0	NULL

1 = show plan for current statement

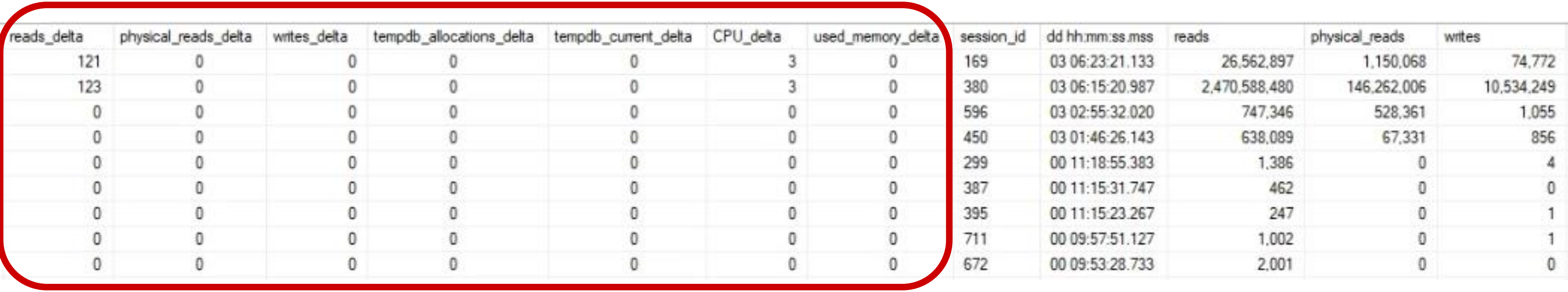
2 = show plan for all queries in the batch

Situational sorcery

What is REALLY going on now?



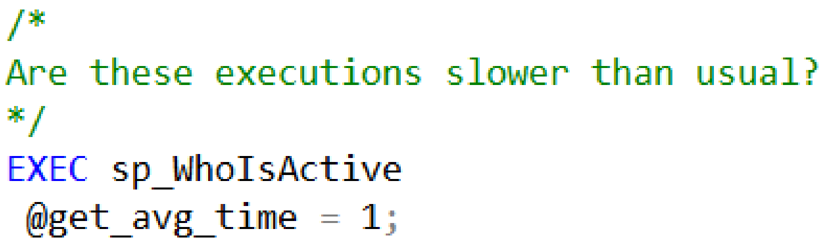
```
/*
Sample activity (delta interval in seconds)
*/
EXEC sp_WhoIsActive
 @output_column_list = '[%delta][%]'
 , @delta_interval = 5;
```



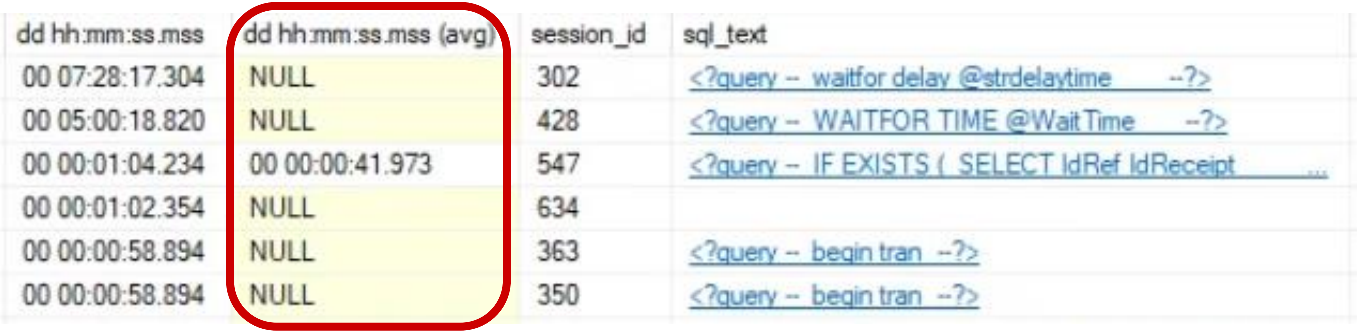
reads_delta	physical_reads_delta	writes_delta	tempdb_allocations_delta	tempdb_current_delta	CPU_delta	used_memory_delta	session_id	dd hh:mm:ss.mss	reads	physical_reads	writes
121	0	0	0	0	3	0	169	03 06:23:21.133	26,562,897	1,150,068	74,772
123	0	0	0	0	3	0	380	03 06:15:20.987	2,470,588,480	146,262,006	10,534,249
0	0	0	0	0	0	0	596	03 02:55:32.020	747,346	528,361	1,055
0	0	0	0	0	0	0	450	03 01:46:26.143	638,089	67,331	856
0	0	0	0	0	0	0	299	00 11:18:55.383	1,386	0	4
0	0	0	0	0	0	0	387	00 11:15:31.747	462	0	0
0	0	0	0	0	0	0	395	00 11:15:23.267	247	0	1
0	0	0	0	0	0	0	711	00 09:57:51.127	1,002	0	1
0	0	0	0	0	0	0	672	00 09:53:28.733	2,001	0	0

Most values indicate number of 8K data pages

Is this abnormal?



```
/*
Are these executions slower than usual?
*/
EXEC sp_WhoIsActive
 @get_avg_time = 1;
```

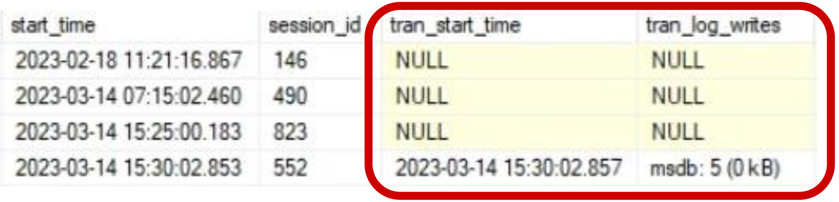


dd hh:mm:ss.mss	dd hh:mm:ss.mss (avg)	session_id	sql_text
00 07:28:17.304	NULL	302	<?query -- waitfor delay @strdelaytime --?>
00 05:00:18.820	NULL	428	<?query -- WAITFOR TIME @WaitTime --?>
00 00:01:04.234	00 00:00:41.973	547	<?query -- IF EXISTS (SELECT IdRef IdReceipt ...
00 00:01:02.354	NULL	634	
00 00:00:58.894	NULL	363	<?query -- begin tran --?>
00 00:00:58.894	NULL	350	<?query -- begin tran --?>

Average time of execution...if available

What filled up the log?

```
/*
What filled up the transaction log?
*/
EXEC sp_WhoIsActive
 @output_column_list = '[start_time][session_id][tran%][login%][sql_text][%]'
 , @get_transaction_info = 1;
```

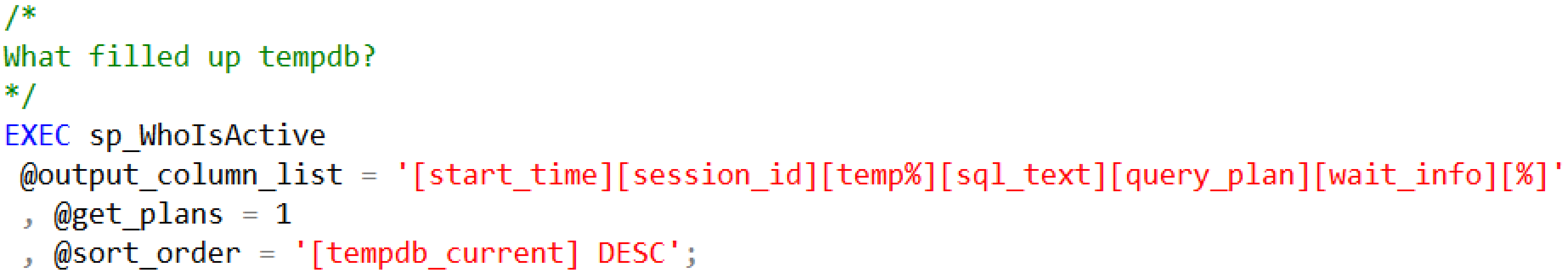


start_time	session_id	tran_start_time	tran_log_writes
2023-02-18 11:21:16.867	146	NULL	NULL
2023-03-14 07:15:02.460	490	NULL	NULL
2023-03-14 15:25:00.183	823	NULL	NULL
2023-03-14 15:30:02.853	552	2023-03-14 15:30:02.857	msdb: 5 (0 kB)

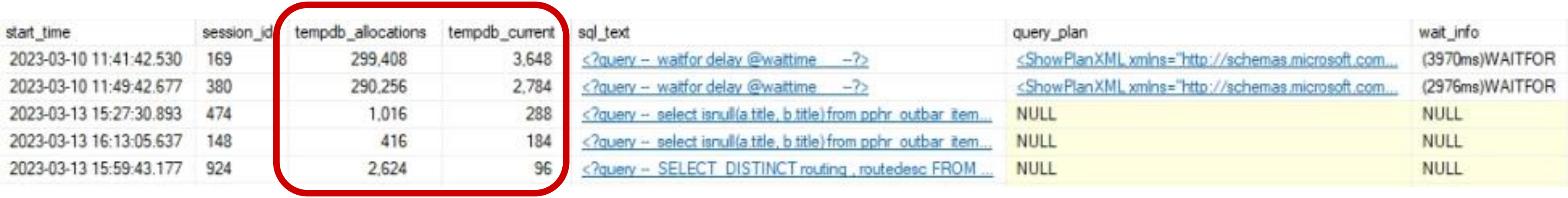
Tran_start_time: date/time first database was written to

Tran_log writes: # of log records (size of log records)

What filled up tempdb?



```
/*
What filled up tempdb?
*/
EXEC sp_WhoIsActive
 @output_column_list = '[start_time][session_id][temp%][sql_text][query_plan][wait_info][%]'
 , @get_plans = 1
 , @sort_order = '[tempdb_current] DESC';
```



start_time	session_id	tempdb_allocations	tempdb_current	sql_text	query_plan	wait_info
2023-03-10 11:41:42.530	169	299,408	3,648	<?query -- waitfor delay @waittime --?>	<ShowPlanXML xmlns="http://schemas.microsoft.com...	(3970ms)WAITFOR
2023-03-10 11:49:42.677	380	290,256	2,784	<?query -- waitfor delay @waittime --?>	<ShowPlanXML xmlns="http://schemas.microsoft.com...	(2976ms)WAITFOR
2023-03-13 15:27:30.893	474	1,016	288	<?query -- select isnull(a.title, b.title) from pphr_outbar_item...	NULL	NULL
2023-03-13 16:13:05.637	148	416	184	<?query -- select isnull(a.title, b.title) from pphr_outbar_item...	NULL	NULL
2023-03-13 15:59:43.177	924	2,624	96	<?query -- SELECT DISTINCT routing , routedesc FROM ...	NULL	NULL

High tempdb_allocations indicates high usage

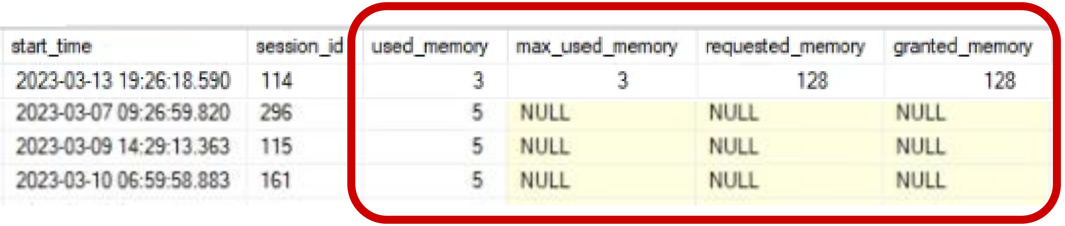
High tempdb_current indicates growth

#PASSDataCommunitySummit

PASS Data Community SUMMIT 2023

What filled up memory?

```
/*
What filled up the memory? (v12 only)
*/
EXEC sp_WhoIsActive
 @output_column_list = '[start_time][session_id][%memory][login%][sql_text][%]'
 , @get_memory_info = 1;
```



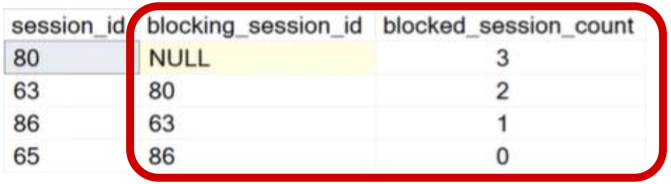
start_time	session_id	used_memory	max_used_memory	requested_memory	granted_memory
2023-03-13 19:26:18.590	114	3	3	128	128
2023-03-07 09:26:59.820	296	5	NULL	NULL	NULL
2023-03-09 14:29:13.363	115	5	NULL	NULL	NULL
2023-03-10 06:59:58.883	161	5	NULL	NULL	NULL

Watch for large grants with low used_memory

What is causing the blocking?

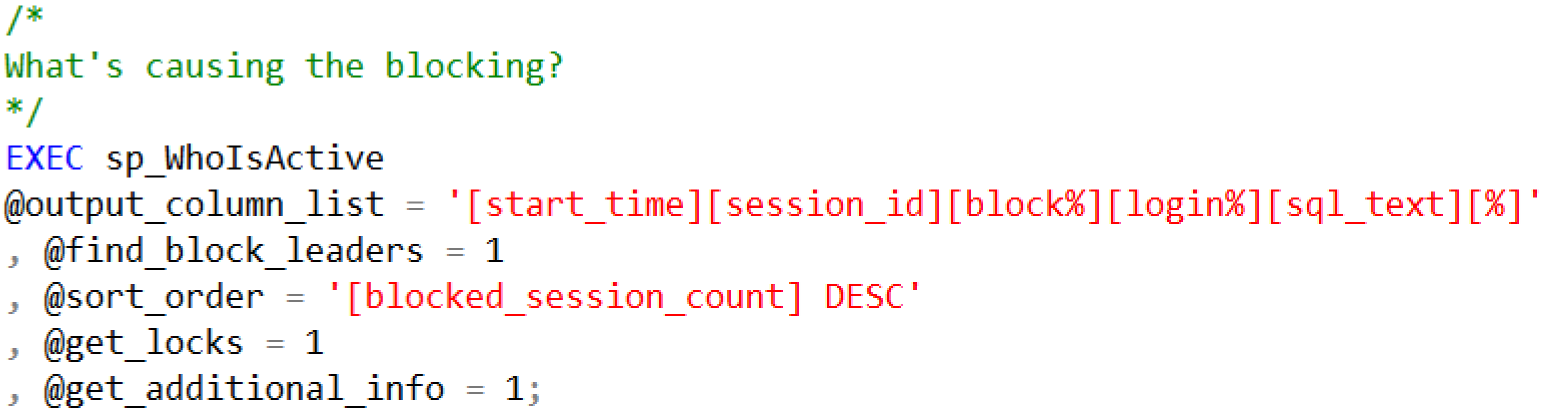
```
/*
What's causing the blocking?
*/
EXEC sp_WhoIsActive
@output_column_list = '[start_time][session_id][block%][login%][sql_text][%]'
, @find_block_leaders = 1
, @sort_order = '[blocked_session_count] DESC'
, @get_locks = 1
, @get_additional_info = 1;
```

@find_block_leaders adds blocked_session_count



session_id	blocking_session_id	blocked_session_count
80	NULL	3
63	80	2
86	63	1
65	86	0

What is causing the blocking?

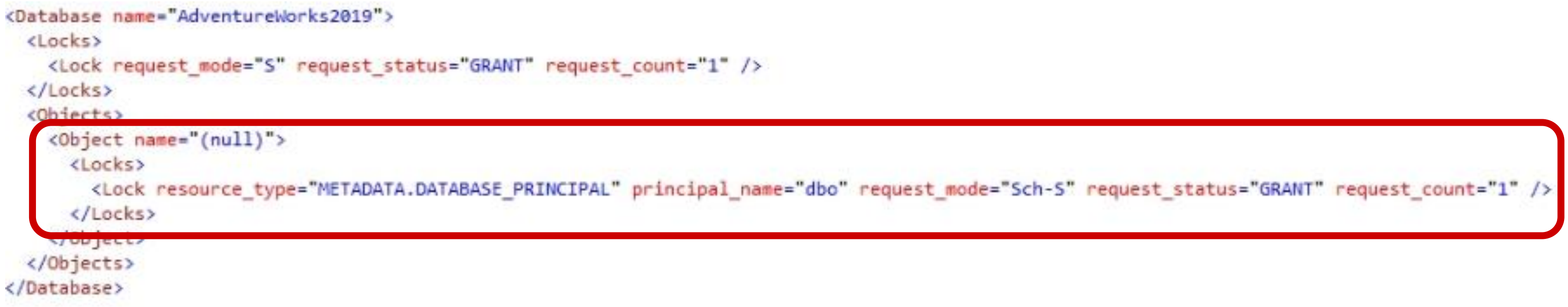


```
/*
What's causing the blocking?
*/
EXEC sp_WhoIsActive
@output_column_list = '[start_time][session_id][block%][login%][sql_text][%]'
, @find_block_leaders = 1
, @sort_order = '[blocked_session_count] DESC'
, @get_locks = 1
, @get_additional_info = 1;
```

@find_block_leaders adds blocked_session_count

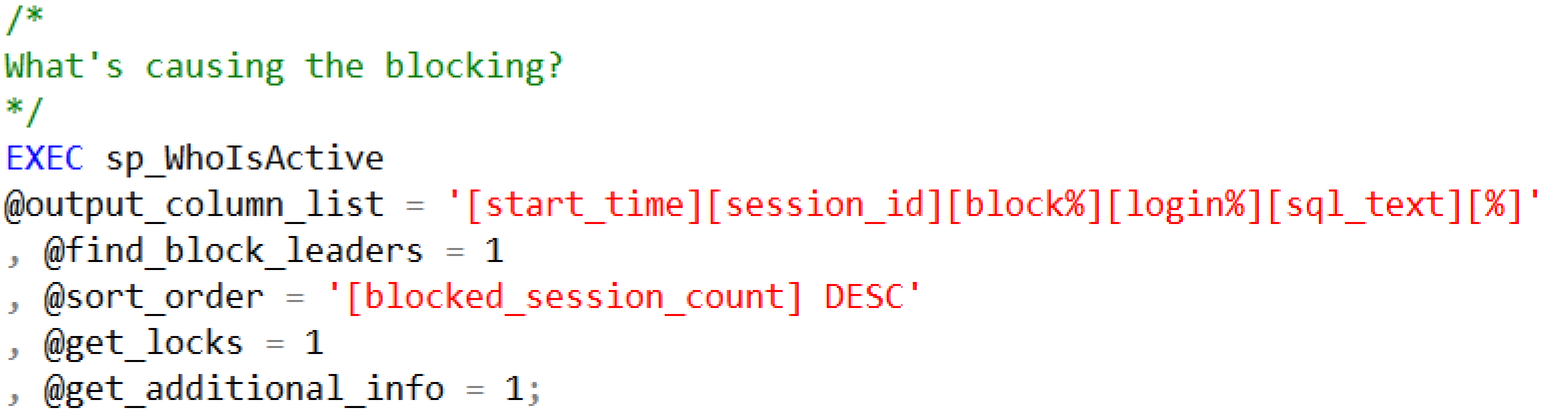
@get_locks adds locks column (XML)

@get_locks



```
<Database name="AdventureWorks2019">
  <Locks>
    <Lock request_mode="S" request_status="GRANT" request_count="1" />
  </Locks>
  <Objects>
    <Object name="(null)">
      <Locks>
        <Lock resource_type="METADATA.DATABASE_PRINCIPAL" principal_name="dbo" request_mode="Sch-S" request_status="GRANT" request_count="1" />
      </Locks>
    </Object>
  </Objects>
</Database>
```

What is causing the blocking?



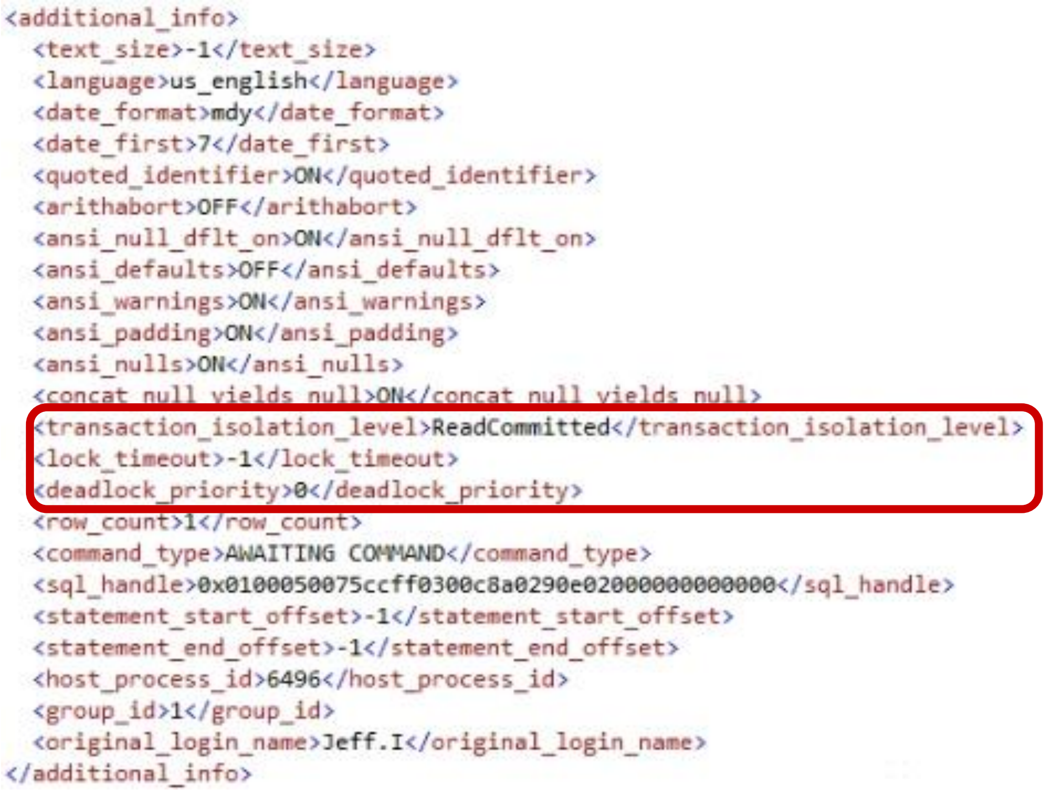
```
/*
What's causing the blocking?
*/
EXEC sp_WhoIsActive
@output_column_list = '[start_time][session_id][block%][login%][sql_text][%]'
, @find_block_leaders = 1
, @sort_order = '[blocked_session_count] DESC'
, @get_locks = 1
, @get_additional_info = 1;
```

@find_block_leaders adds blocked_session_count

@get_locks adds locks (XML)

@get_additional_info adds additional_info (XML)

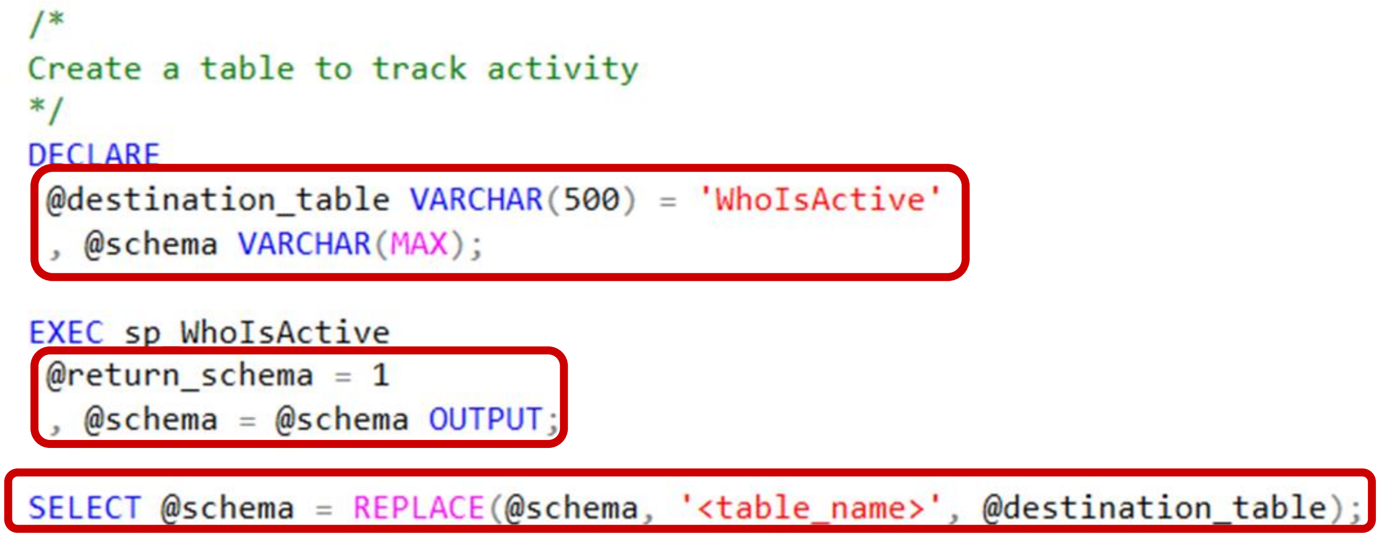
@get_additional_info



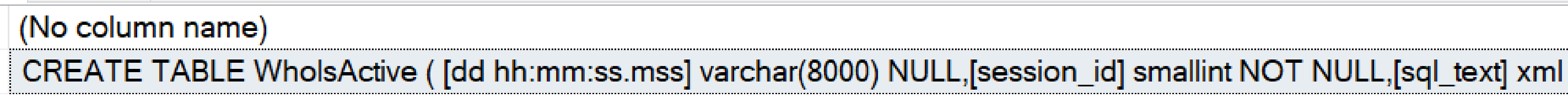
```
<additional_info>
  <text_size>-1</text_size>
  <language>us_english</language>
  <date_format>mdy</date_format>
  <date_first>7</date_first>
  <quoted_identifier>ON</quoted_identifier>
  <arithabort>OFF</arithabort>
  <ansi_null_dflt_on>ON</ansi_null_dflt_on>
  <ansi_defaults>OFF</ansi_defaults>
  <ansi_warnings>ON</ansi_warnings>
  <ansi_padding>ON</ansi_padding>
  <ansi_nulls>ON</ansi_nulls>
  <concat_null_yields_null>ON</concat_null_yields_null>
  <transaction_isolation_level>ReadCommitted</transaction_isolation_level>
  <lock_timeout>-1</lock_timeout>
  <deadlock_priority>0</deadlock_priority>
  <row_count>1</row_count>
  <command_type>AWAITING COMMAND</command_type>
  <sql_handle>0x0100050075ccff0300c8a0290e02000000000000</sql_handle>
  <statement_start_offset>-1</statement_start_offset>
  <statement_end_offset>-1</statement_end_offset>
  <host_process_id>6496</host_process_id>
  <group_id>1</group_id>
  <original_login_name>Jeff.I</original_login_name>
</additional_info>
```

Time travelling

Create a table

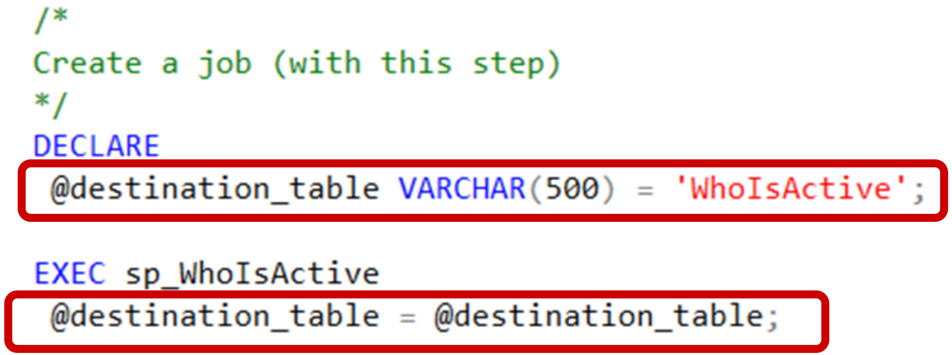


```
/*
Create a table to track activity
*/
DECLARE
 @destination_table VARCHAR(500) = 'WhoIsActive'
 , @schema VARCHAR(MAX);

EXEC sp_WhoIsActive
 @return_schema = 1
 , @schema = @schema OUTPUT;

SELECT @schema = REPLACE(@schema, '<table_name>', @destination_table);
```



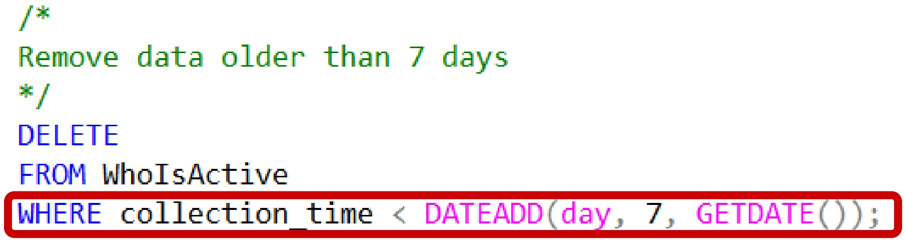
```
(No column name)
CREATE TABLE WhoIsActive ( [dd hh:mm:ss.mss] varchar(8000) NULL,[session_id] smallint NOT NULL,[sql_text] xml
```

Create a job



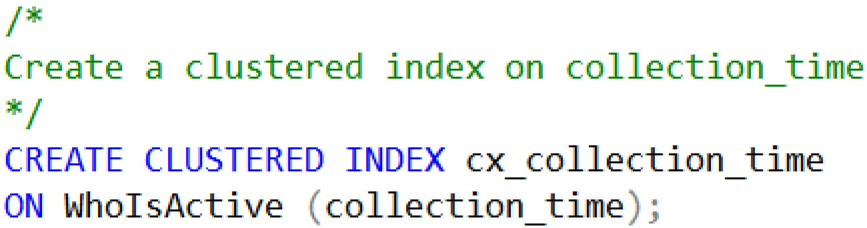
```
/*
Create a job (with this step)
*/
DECLARE
 @destination_table VARCHAR(500) = 'WhoIsActive';

EXEC sp_WhoIsActive
 @destination_table = @destination_table;
```

Add a cleanup step



```
/*
Remove data older than 7 days
*/
DELETE
FROM WhoIsActive
WHERE collection_time < DATEADD(day, 7, GETDATE());
```

Create an index

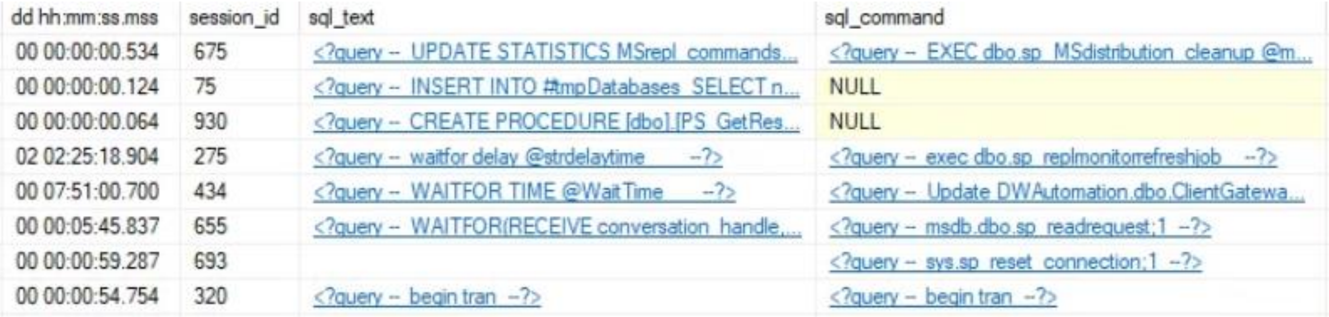


```sql
/*
Create a clustered index on collection_time
*/
CREATE CLUSTERED INDEX cx_collection_time
ON WhoIsActive (collection_time);
```

Optional...but highly recommended

Query the table

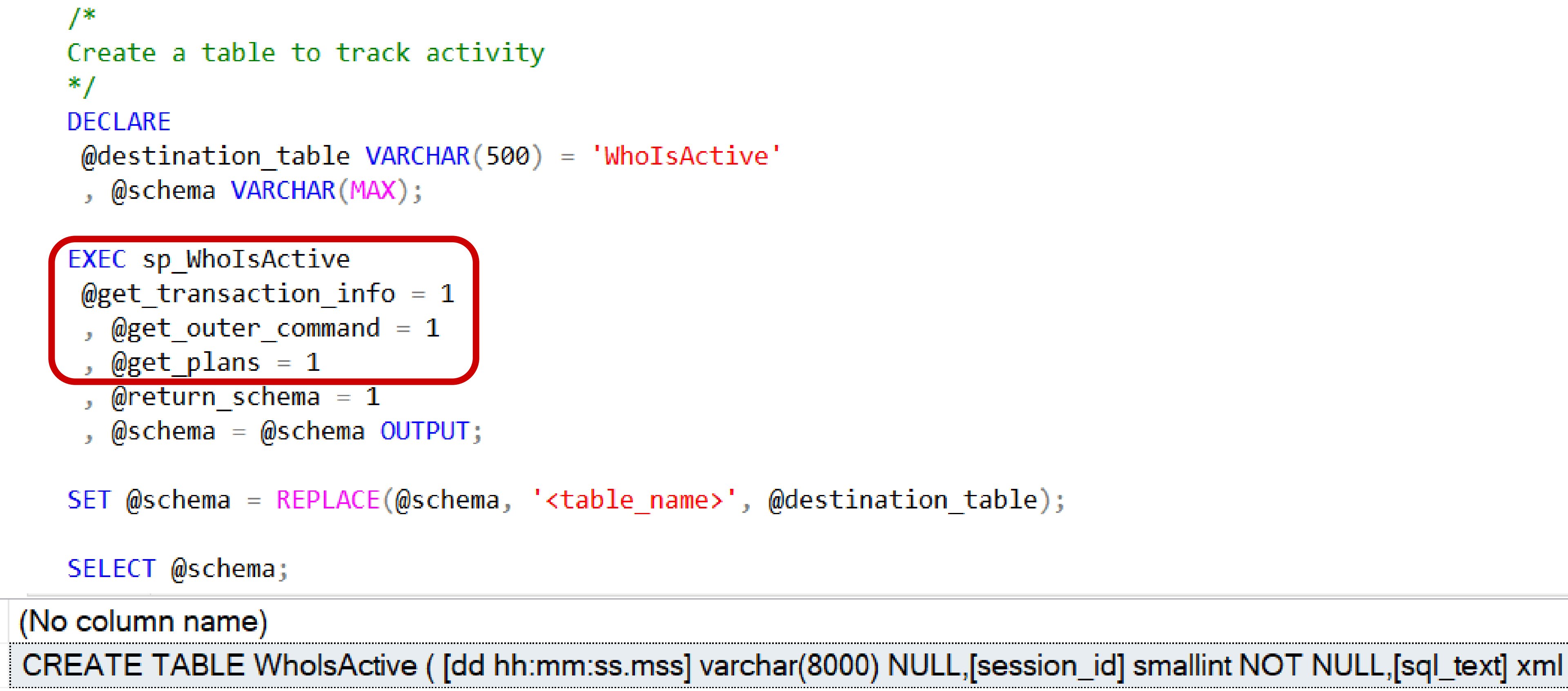
```
/*
Query the table
*/
SELECT *
FROM WhoIsActive
WHERE collection_time BETWEEN '2023-11-15 00:00:00' AND '2023-11-15 01:00:00';
```



dd hh:mm:ss.mss	session_id	sql_text	sql_command
00 00:00:00.534	675	<?query -- UPDATE STATISTICS MSrepl_commands...	<?query -- EXEC dbo.sp_MSdistribution_cleanup @m...
00 00:00:00.124	75	<?query -- INSERT INTO #tmpDatabases SELECT n...	NULL
00 00:00:00.064	930	<?query -- CREATE PROCEDURE [dbo].[PS_GetRes...	NULL
02 02:25:18.904	275	<?query -- waitfor delay @strdelaytime --?>	<?query -- exec dbo.sp_replmonitorrefreshjob --?>
00 07:51:00.700	434	<?query -- WAITFOR TIME @WaitTime --?>	<?query -- Update DWAutomation.dbo.ClientGatewa...
00 00:05:45.837	655	<?query -- WAITFOR(RECEIVE conversation_handle,...	<?query -- msdb.dbo.sp_readrequest;1 --?>
00 00:00:59.287	693		<?query -- sys.sp_reset_connection;1 --?>
00 00:00:54.754	320	<?query -- begin tran --?>	<?query -- begin tran --?>

#PASSDataCommunitySummit

Customize your output



```
/*
Create a table to track activity
*/
DECLARE
 @destination_table VARCHAR(500) = 'WhoIsActive'
  , @schema VARCHAR(MAX);

EXEC sp_WhoIsActive
 @get_transaction_info = 1
  , @get_outer_command = 1
  , @get_plans = 1
  , @return_schema = 1
  , @schema = @schema OUTPUT;

SET @schema = REPLACE(@schema, '<table_name>', @destination_table);

SELECT @schema;
```

```
(No column name)
CREATE TABLE WhoIsActive ( [dd hh:mm:ss.mss] varchar(8000) NULL,[session_id] smallint NOT NULL,[sql_text] xml
```

Send yourself alerts

Long-running queries

Large memory grants

Tempdb usage

Blocking

PASS
Data Community
SUMMIT
2023

It’s demo time!

Summary – the basics

@output_column_list

@filter_type	@filter
@not_filter_type	@not_filter_type

@sort_order

Summary – hidden gems

@show_sleeping_spids

@show_system_spids

@get_outer_command

@get_plans

Summary – situational sorcery

- @delta_interval
- @get_avg_time
- @get_transaction_info
- @get_memory_info
- @find_block_leaders
- @get_locks
- @get_additional_info

Summary – time travelling

@destination_table

@return_schema

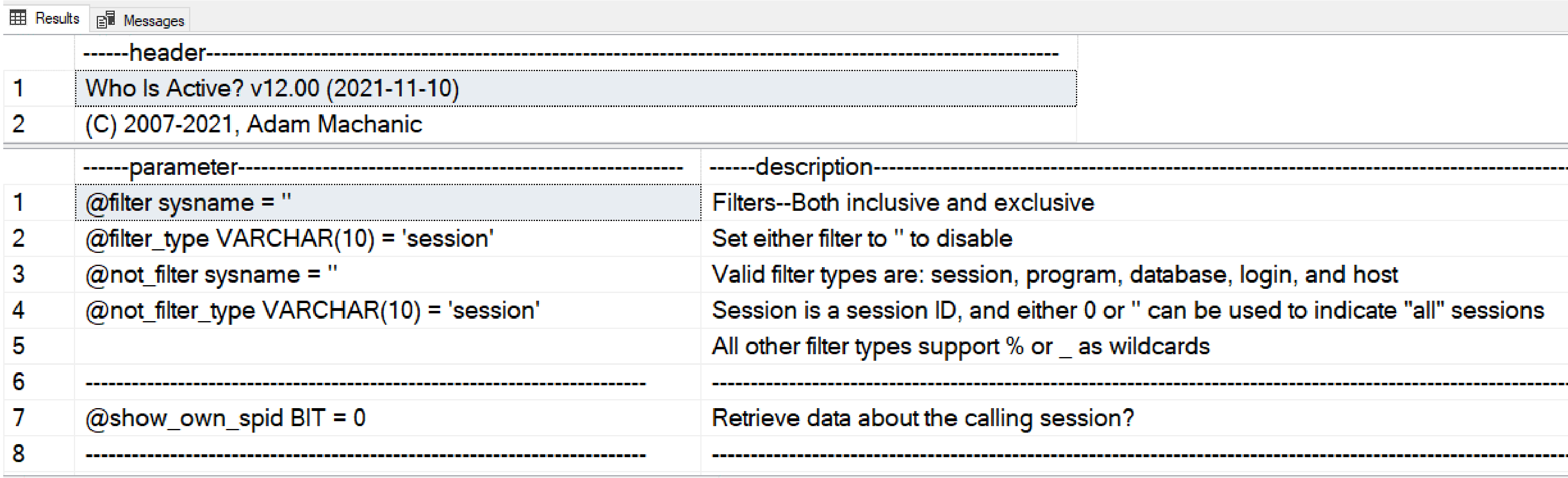
@schema

We’re all wizards now (but in case we’re not…)

PASS Data Community SUMMIT 2023

@help

```sql
/*
Get the documentation
*/
EXEC sp_WhoIsActive
 @help = 1
```



	------header--
1	Who Is Active? v12.00 (2021-11-10)
2	(C) 2007-2021, Adam Machanic

	------parameter--	------description--
1	@filter sysname = ''	Filters--Both inclusive and exclusive
2	@filter_type VARCHAR(10) = 'session'	Set either filter to '' to disable
3	@not_filter sysname = ''	Valid filter types are: session, program, database, login, and host
4	@not_filter_type VARCHAR(10) = 'session'	Session is a session ID, and either 0 or '' can be used to indicate "all" sessions
5		All other filter types support % or _ as wildcards
6	--	--
7	@show_own_spid BIT = 0	Retrieve data about the calling session?
8	--	--

Session evaluation

Your feedback is important to us

Evaluate this session at:

www.PASSDataCommunitySummit.com/evaluation

PASS
Data Community
SUMMIT
2023

Thank you

Unlocking the Power of WhoIsActive

Jeff Iannucci

github.com\desertdba

jeff@desertdba.com

www.linkedin.com/in/jeff-iannucci

PASS
Data Community
SUMMIT
2023