



Advanced query signature view

SQL Diagnostic Manager provides the ability to view individual SQL Statements or to view Query Signatures. Query signatures are groupings of SQL Statements that match once their literals are stripped. A query signature broadly defines queries and trends with a less overwhelming amount of data and allows you to diagnose a query in a general sense.

Once you identify a potential problem signature, you can drill into individual queries that make up the signature as needed by clicking the  icon.

QUERY SIGNATURE (2)												Options	Show Top 100 200 All
Query ID	Signature SQL Text	Occurrences	Event Type	Total Duration (s)	Total CPU Time (ms)	Total Reads	Total Writes	Avg Wait Time (ms)	Most Recent Completion	Avg Blocking Time (ms)	Avg Deadlock Time (ms)		
Query Signature 1 -- SQLInventory...			Stored Proc...	113,340	189	7,424	78	0.00	09/06/16 02:51 A	0.00	0.00		
Query Signature 2 -- SQLInventory...			SQL Statemen...	1,653,487	2,291	147,285	203	0.00	09/07/16 10:24 A	0.00	0.00		

Upon clicking the  icon, the following context menu displays:

- **View SQL.** Click this option to view the underlying SQL Text for the query execution plan. In this view, you can export and save SQL text for review

Query Details

SQL Text

Export

```

1  -- SQLInventoryManager -- Copyright © 2013-2014-2016, Idera, Inc., All
2  Rights Reserved. set transaction isolation level read
3  uncommitted; set lock_timeout #; set implicit_transactions off;
4  if @@trancount > # commit transaction; set language us_english;
5  set cursor_close_on_commit off; set query_governor_cost_limit
6  #; set numeric_roundabort off; set deadlock_priority low; set
7  nocount on; use master;
8
9  if object_id(#) is not null drop table #db_files;
10
11 create table #db_files( [db_id] int, [file_id] int, [type] tinyint,
12 [page_count] bigint, [pages_used] bigint ) declare @cmd
13 nvarchar(#) declare @db_name nvarchar(#) declare db_cursor
14 cursor static read_only for select name from sys.databases
15 where HAS_DBACCESS(name)=# order by database_id open db_cursor
16 fetch next from db_cursor into @db_name while @@FETCH_STATUS =
17 # begin set @cmd = # + replace(@db_name,#,#) + # + # + # + # + #
18 + # begin try insert into #db_files execute (@cmd) end try begin
19 catch -- try sys.master_files to get size but not space used
20 set @cmd = #+@db_name+# + # + # + # + # + #+@db_name+# begin try
21 insert into #db_files execute (@cmd) end try begin catch -- add
22 entry so we know about database insert into #db_files values
23 (DB_ID(@db_name), #, #, NULL, NULL) end catch end catch fetch
24 next from db_cursor into @db_name end close db_cursor
25 deallocate db_cursor select d.name, db.* from #db_files db join
26 sys.databases d on d.database_id = db.db_id drop table
27 #db_files

```

Close

- **Query Statement.** Click this option to view query statements for the selected client.

For additional information on query signatures, see [View the query monitor signature mode](#).

SQL Diagnostics Manager for SQL Server performance monitoring, alerting, and diagnostics for SQL Server.

[IDERA](#) | [Products](#) | [Purchase](#) | [Support](#) | [Community](#) | [Resources](#) | [About Us](#) | [Legal](#)