sp_lock (Transact-SQL)
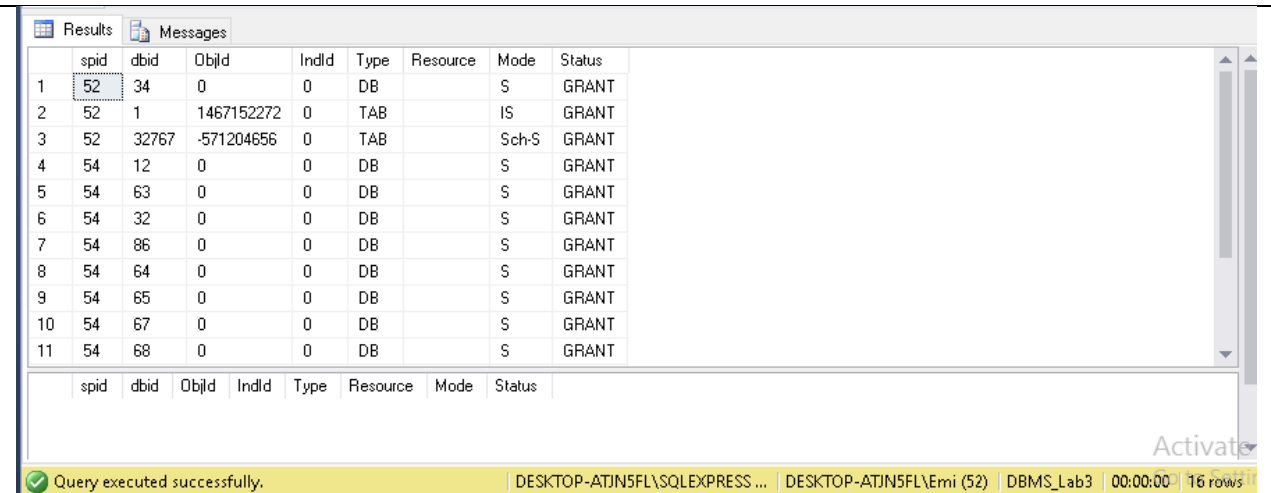
The **sp_lock** result set contains one row for each lock held by the sessions specified in the **@spid1** and **@spid2** parameters. If neither **@spid1** nor **@spid2** is specified, the result set reports the locks for all sessions currently active in the instance of the Database Engine.

```
use DBMS_Lab3
go

-- sp_lock

-- A.  Listing all locks -  displays information about all locks currently held in an
instance of the Database Engine.
EXEC sp_lock;
GO

-- B.  Listing a lock from a single-server process -  displays information, including
locks, about process ID 53.
EXEC sp_lock 53;
GO
```

	spid	dbid	ObjId	IndId	Type	Resource	Mode	Status
1	52	34	0	0	DB		S	GRANT
2	52	1	1467152272	0	TAB		IS	GRANT
3	52	32767	-571204656	0	TAB		Sch-S	GRANT
4	54	12	0	0	DB		S	GRANT
5	54	63	0	0	DB		S	GRANT
6	54	32	0	0	DB		S	GRANT
7	54	86	0	0	DB		S	GRANT
8	54	64	0	0	DB		S	GRANT
9	54	65	0	0	DB		S	GRANT
10	54	67	0	0	DB		S	GRANT
11	54	68	0	0	DB		S	GRANT

| spid | dbid | ObjId | IndId | Type | Resource | Mode | Status |

Query executed successfully.   DESKTOP-ATJN5FL\SQLEXPRESS ...   DESKTOP-ATJN5FL\Emi (52)   DBMS_Lab3   00:00:00   16 rows

sys.dm_tran_locks (Transact-SQL)

Returns information about currently active lock manager resources in SQL Server. Each row represents a currently active request to the lock manager for a lock that has been granted or is waiting to be granted. The columns in the result set are divided into two main groups: resource and request. The resource group describes the resource on which the lock request is being made, and the request group describes the lock request.

```
-- sys.dm_tran_locks

--     The following query will display lock information. The value for <dbid> should be
replaced with the database_id from sys.databases.

select * from sys.databases
-- the database_id of my database DBMS_Lab3 is 34
```

```
SELECT resource_type, resource_associated_entity_id,
    request_status, request_mode,request_session_id,
    resource_description
    FROM sys.dm_tran_locks
    WHERE resource_database_id = 34 -- <dbid>

--    The following query will show blocking information.
SELECT
        t1.resource_type,
        t1.resource_database_id,
        t1.resource_associated_entity_id,
        t1.request_mode,
        t1.request_session_id,
        t2.blocking_session_id
    FROM sys.dm_tran_locks as t1
    INNER JOIN sys.dm_os_waiting_tasks as t2
        ON t1.lock_owner_address = t2.resource_address;
```

Results | Messages

	resource_type	resource_associated_entity_id	request_status	request_mode	request_session_id	resource_description
1	DATABASE	0		GRANT	S	52	

resource_type	resource_database_id	resource_associated_entity_id	request_mode	request_session_id	blocking_session_id

Query executed successfully. | DESKTOP-ATJN5FL\SQLEXPRESS ... | DESKTOP-ATJN5FL\Emi (52) | DBMS_Lab

sys.dm_tran_active_transactions (Transact-SQL)

Returns information about transactions for the instance of SQL Server.

SET QUERY_GOVERNOR_COST_LIMIT

SET QUERY_GOVERNOR_COST_LIMIT value

Value - Is a numeric or integer value specifying the longest time in which a query can run. Values are rounded down to the nearest integer. Negative values are rounded up to 0. The query governor disallows execution of any query that has an estimated cost exceeding that value. Specifying 0 (the default) for this option turns off the query governor, and all queries are allowed to run indefinitely.

"Query cost" refers to the estimated elapsed time, in seconds, required to complete a query on a specific hardware configuration.

Using SET QUERY_GOVERNOR_COST_LIMIT applies to the current connection only and lasts the duration of the current connection. Use the Configure the query governor cost limit

Server Configuration Optionoption of **sp_configure** to change the server-wide query governor cost limit value.

The setting of SET QUERY_GOVERNOR_COST_LIMIT is set at execute or run time and not at parse time.

```
-- SET QUERY_GOVERNOR_COST_LIMIT

-- how to use sp_configure to set the value of the query governor cost limit option to
120 seconds.
USE DBMS_Lab3 ;
GO
EXEC sp_configure 'show advanced options', 1;
GO
RECONFIGURE ;
GO
EXEC sp_configure 'query governor cost limit', 120 ;
GO
RECONFIGURE;
GO
-- The setting takes effect immediately without restarting the server.
```

Messages

    Configuration option 'show advanced options' changed from 1 to 1. Run the RECONFIGURE statement to install.
    Configuration option 'query governor cost limit' changed from 0 to 120. Run the RECONFIGURE statement to install.

DBCC Commands = DataBase Control Commands for SQL Server

- DBCC DBTABLE – this command displays the system information for the specified database's tables. The command takes a single parameter that specifies the database name.

      DBCC DBTABLE(pubs)

- DBCC DBINFO - this command returns the database information structure for the specified database. The command takes one parameter, which identifies the database.

      DBCC DBINFO(pubs)

- DBCC PROCBUF – this command displays the contents of the SQL Server Procedure Buffer, a storage area where SQL Server caches executable statements such as stored procedures and SQL queries. This command shows procedure and buffer-header information. The parameters specify the database, the stored procedure, the buffer number, and the print option, in that order.

      DBCC PROCBUF('master', 'sp_help', 1, 0)

- DBCC BUFFER - this command displays the buffer headers and pages from SQL Server's buffer cache, where SQL Server stores results. The parameters specify the database name and the object name.

DBCC BUFFER(pubs, 'sysobjects')

- DBCC SHOWFILESTATS - this statement returns space usage information about all the data files in the active database. The command returns one record for each data file used in the database.

DBCC SHOWFILESTATS

- DBCC PAGE - this command shows the data page structure that a given database uses. The first parameter specifies the database name, the second parameter specifies the page number, and the third parameter is the print option.

DBCC PAGE('pubs', 1, 1)

- DBCC LOG- this command shows the transaction log for the specified database. The DBCC LOG command's first parameter is the database name, and the second parameter specifies the level of information to show to the user.

DBCC LOG('pubs', 3)

Level	What Does?
0	Return only the minimum of information for each operation -- the operation, its context and the transaction ID. (Default)
1	As 0, but also retrieve any flags and the log record length.
2	As 1, but also retrieve the object name, index name, page ID and slot ID.
3	Full informational dump of each operation.
4	As 3 but includes a hex dump of the current transaction log row.

Example:

References:

https://searchsqlserver.techtarget.com/tip/View-SQL-Server-transaction-logs-using-DBCC
http://www.itprotoday.com/microsoft-sql-server/dbcc-commands

https://www.sqlshack.com/useful-dbcc-log-commands/

https://docs.microsoft.com/en-us/sql/t-sql/statements/set-query-governor-cost-limit-transact-sql?view=sql-server-2017

https://docs.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/sp-configure-transact-sql?view=sql-server-2017

https://docs.microsoft.com/en-us/sql/relational-databases/system-dynamic-management-views/sys-dm-tran-locks-transact-sql?view=sql-server-2017

https://docs.microsoft.com/en-us/sql/relational-databases/system-dynamic-management-views/sys-dm-tran-active-transactions-transact-sql?view=sql-server-2017

http://www.sqlservercentral.com/articles/Administration/sqlserver7someusefulundocumenteddbc ccommands/405/#part_2_8