Write a query and put the script file on hard disk Ex: D:\Scripts\BlockingQuery.sql

sp\_readerrorlog

selectcmd,\*fromsys.sysprocesses

whereblocked> 0

usemaster;

select p.spid, t.text

fromsysprocesses p

CROSSAPPLYsys.dm\_exec\_sql\_text(sql\_handle) t

where p.blocked = 0

and p.spid in

(select p1.blocked

fromsysprocesses p1

where p1.blocked > 0

and p1.waittime > 50 )

selectloginame,cpu,memusage,physical\_io,\*

frommaster..sysprocessesa

whereexists(selectb.\*

frommaster..sysprocessesb

whereb.blocked> 0 and

b.blocked=a.spid)andnot

exists(selectb.\*

frommaster..sysprocessesb

whereb.blocked> 0 and

b.spid=a.spid)

orderbyspid

USE Master

GO

SELECT blocking\_session\_id, \*

FROM sys.dm\_exec\_requests

WHERE blocking\_session\_id <> 0;

GO

USE Master

GO

SELECT session\_id, wait\_duration\_ms, wait\_type, blocking\_session\_id

FROM sys.dm\_os\_waiting\_tasks

WHERE blocking\_session\_id <> 0

order by wait\_duration\_ms desc

------------------------------------------------------------------------

SELECT session\_id

,blocking\_session\_id

,wait\_time

,wait\_type

,last\_wait\_type

,wait\_resource

,transaction\_isolation\_level

,lock\_timeout

FROM sys.dm\_exec\_requests

WHERE blocking\_session\_id<> 0

GO

From command prompt run the script on sql server and get the result to a text file

SQLCMD -A – SMyServer -i”C:\SQLScripts\GetBlockers.sql” -o”C:\SQLScripts\blockers.txt”

**SQL: View missing Indexes using SQL Sever DMVs**

Sys.dm\_db\_missing\_index\_details

Sys.dm\_db\_missing\_index\_group\_stats

Sys.dm\_db\_missing\_index\_group

Sp\_spaceused <tablename>

Sp\_helpindex <tablename>

Sys.dm\_db\_index\_physical\_stats

SELECT TOP 20

S.text AS [SQL]

, P.cacheobjtype

, P.objtype

, DB\_NAME(S.dbid)AS [DatabaseName]

, P.usecounts AS [Plan usage]

, Q.query\_plan

FROM sys.dm\_exec\_cached\_plans P

CROSS APPLY sys.dm\_exec\_sql\_text(P.plan\_handle) S

CROSS APPLY sys.dm\_exec\_query\_plan(P.plan\_handle) Q

WHERE

CAST(Q.query\_plan AS NVARCHAR(MAX))LIKE '%<MissingIndexes>%'

ORDER BY P.usecounts DESC

[Determine the size of data stored in a Column](http://www.mrtsql.com/2012/08/determine-size-of-data-stored-in-column.html)

**SELECT DATALENGTH(MyColumn) AS SizeInBytes FROM Table**

# [SQL Change Data Capture - Automatically generate DML UNDO/REDO Scripts](http://www.mrtsql.com/2011/11/sql-change-data-capture-automatically.html)

USE <Database>

GO

IF (SELECT is\_cdc\_enabled FROM sys.databases WHERE name=db\_name())=0

         exec sys.sp\_cdc\_enable\_db

SELECT 'exec sys.sp\_cdc\_enable\_table @source\_schema = ''' + SCHEMA\_NAME([Schema\_id]) + ''', @source\_name = '''+ name +''',@role\_name = ''CDCRole'',@supports\_net\_changes = 0, @capture\_instance='''+ name +'\_DEPLOY''' AS [Command] FROM sys.tables WHERE SCHEMA\_NAME([Schema\_id]) !='cdc' AND name != 'systranschemas' ORDER BY [name]

# [How to rename a SQL Instance](http://www.mrtsql.com/2013/01/how-to-rename-sql-instance.html)

DECLARE     @OldServerName VARCHAR(50),

            @NewServerName VARCHAR(50)

SELECT @NewServerName='<New Server Name>'

SELECT @OldServerName=srvname FROM master.dbo.sysservers WHERE srvid=0

EXEC sp\_dropserver @OldServerName

EXEC sp\_addserver @NewServerName, local

# [Return Deadlock information](http://www.mrtsql.com/2012/11/sql-return-deadlock-information.html)

SELECT

XEventData.XEvent.value('(data/value)[1]','varchar(max)')ASDeadlockGraph

FROM

(SELECTCAST(target\_dataASXML)ASTargetData

FROMsys.dm\_xe\_session\_targetsst

JOINsys.dm\_xe\_sessionss

ONs.address=st.event\_session\_address

WHEREname='system\_health')ASData

CROSSAPPLYTargetData.nodes('//RingBufferTarget/event')ASXEventData(XEvent)

WHERE

XEventData.XEvent.value('@name','varchar(4000)')='xml\_deadlock\_report'

**Change the tempdb into another drive**

USE master

GO

ALTER DATABASE TempDB MODIFY FILE

(NAME = tempdev, FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL.3\MSSQL\DATA\tempdb.mdf')

GO

ALTER DATABASE TempDB MODIFY FILE

(NAME = templog, FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL.3\MSSQL\DATA\templog.ldf')

GO

SELECT name, physical\_name AS CurrentLocation, state\_desc

FROM sys.master\_files

WHERE database\_id = DB\_ID('master');

GO

SELECT name, physical\_name AS CurrentLocation

FROM sys.master\_files

WHERE database\_id = DB\_ID(N'tempdb');

GO

If we have multiple tempdb:

SELECT 'ALTER DATABASE tempdb MODIFY FILE (NAME = [' + f.name + '],'

+ ' FILENAME = ''Z:\MSSQL\DATA\' + f.name

+ CASE WHEN f.type = 1 THEN '.ldf' ELSE '.mdf' END

+ ''');'

FROM sys.master\_files f

WHERE f.database\_id = DB\_ID(N'tempdb');

ALTER DATABASE SPNDOANT0001 SET OFFLINE WITH NO\_WAIT

ALTER DATABASE SPNDOANT0001 SET ONLINE WITH NO\_WAIT

SET XACT\_ABORT on EXEC [USP\_Creditbureau\_Posting] @trans\_dt='2014/11/01'

SET XACT\_ABORT on EXEC [USP\_Creditbureau\_Posting\_Second] @trans\_dt='2014/10/25'

alter table dbo.Branch\_Centre\_Format add Slno int identity (1,1)

SELECT Branch\_Code,Branch\_Name,MAX(Week\_start\_date),MAX(Week\_end\_date) FROM Week\_day\_Process

 WHERE Week\_status is NOT null group by Branch\_Code,Branch\_Name

**Deletion of duplicate Records**

select loan\_no,count(application\_id\_account\_no) from DO\_TBL\_Customer\_Information where loan\_no is notnull and loan\_no <> '' group by loan\_no having count(\*) > 1

select application\_id\_account\_no,count(\*) from DO\_TBL\_Customer\_Information group byapplication\_id\_account\_no having count(\*) > 1

---------------------------------------

WITH SAMPLE\_CTE

AS

(

SELECT \*, ROW\_NUMBER()OVER (PARTITION BY SalesOrderno, ItemNo ORDER BY SalesOrderno, ItemNo)

AS ROW\_NO from dbo.SalesOrderDetails

)

SELECT \* FROM SAMPLE\_CTE;

--------------------------------------------------------------------

WITH SAMPLE\_CTE AS

 (

 SELECT\*,

ROW\_NUMBER()OVER (PARTITION BY SalesOrderno, ItemNo ORDER BY SalesOrderno,ItemNo)

AS ROW\_NO

from dbo.SalesOrderDetails

)

--DELETE FROM SAMPLE\_CTE WHERE ROW\_NO > 1;

------------------------------------------------------------------------

INSERT INTO [SPANDANA-SMA-1].Final.dbo.Account\_source

SELECT \* FROM dbo.Account\_source

INSERT INTO [SPANDANA-SMA-1].Final.dbo.Address\_source

SELECT \* FROM dbo.Address\_source

INSERT INTO [SPANDANA-SMA-1].Final.dbo.Member\_source

SELECT \* FROM dbo.Member\_source

INSERT INTO [SPANDANA-SMA-1].Final.dbo.Branch\_Centre\_Format

SELECT \* FROM dbo.Branch\_Centre\_Format

select count(\*) from account\_source with (nolock)

select count(\*) from address\_source with (nolock)

select count(\*) from member\_source with (nolock)

update dbo.MEMBER\_source set OTHER\_ID\_TYPE\_1\_DESC=convert(char(20),OTHER\_ID\_TYPE\_1\_DESC)

update member\_source set ration\_card= REPLACE(ration\_card, char(39), '') where ration\_card like '%' + char(39) + '%'

# Shrinking Truncate Log File – Log Full

# USE DatabaseName GO DBCC SHRINKFILE(<TransactionLogName>, 0) BACKUP LOG <DatabaseName> WITH TRUNCATE\_ONLY DBCC SHRINKFILE(<TransactionLogName>, 0) GO

# USE [master] GO ALTER DATABASE [TestDb] SET RECOVERY SIMPLE WITH NO\_WAIT DBCC SHRINKFILE(TestDbLog, 1) ALTER DATABASE [TestDb] SET RECOVERY FULL WITH NO\_WAIT GO

# Drive space query

SELECT

[file\_id] AS [File ID],

[type] AS [File Type],

substring([physical\_name],1,1)AS [Drive],

[name] AS [Logical Name],

[physical\_name] AS [Physical Name],

CAST([size] asDECIMAL(38,0))/128. AS [File Size MB],

CAST(FILEPROPERTY([name],'SpaceUsed')ASDECIMAL(38,0))/128. AS [Space Used MB],

(CAST([size] ASDECIMAL(38,0))/128)-(CAST(FILEPROPERTY([name],'SpaceUsed')ASDECIMAL(38,0))/128.)AS [Free Space],

[max\_size] AS [Max Size],

[is\_percent\_growth] AS [Percent Growth Enabled],

[growth] AS [Growth Rate],

SYSDATETIME()AS [Current Date]

FROMsys.database\_files;

===================================

SELECT

SUBSTRING(a.FILENAME, 1, 1) Drive,

[FILE\_SIZE\_MB] =convert(decimal(12,2),

round(a.size/128.000,2)),

[SPACE\_USED\_MB] =convert(decimal(12,2),

round(fileproperty(a.name,'SpaceUsed')/128.000,2)),

[FREE\_SPACE\_MB] =convert(decimal(12,2),

round((a.size-fileproperty(a.name,'SpaceUsed'))/128.000,2)),

[FREE\_SPACE\_%] =convert(decimal(12,2),

(convert(decimal(12,2),round((a.size-fileproperty(a.name,'SpaceUsed'))/128.000,2))

/convert(decimal(12,2),round(a.size/128.000,2))\* 100)),

a.NAME, a.FILENAME

FROM dbo.sysfiles a

ORDERBY Drive, [Name]

------xp\_fixeddrives

------dbcc sqlperf(logspace)

use"MY\_DB"

exec sp\_spaceused

DBCC DROPCLEANBUFFERS

**Tables are using more space**

SELECT

t.NAME AS TableName,

i.name AS indexName,

SUM(p.rows)AS RowCounts,

SUM(a.total\_pages)AS TotalPages,

SUM(a.used\_pages)AS UsedPages,

SUM(a.data\_pages)AS DataPages,

(SUM(a.total\_pages)\* 8)/ 1024 AS TotalSpaceMB,

(SUM(a.used\_pages)\* 8)/ 1024 AS UsedSpaceMB,

(SUM(a.data\_pages)\* 8)/ 1024 AS DataSpaceMB

FROM

sys.tables t

INNERJOIN

sys.indexes i ON t.OBJECT\_ID= i.object\_id

INNERJOIN

sys.partitions p ON i.object\_id= p.OBJECT\_IDAND i.index\_id = p.index\_id

INNERJOIN

sys.allocation\_units a ON p.partition\_id= a.container\_id

WHERE

t.NAME NOTLIKE'dt%'AND

i.OBJECT\_ID> 255 AND

i.index\_id <= 1

GROUPBY

t.NAME, i.object\_id, i.index\_id, i.name

ORDERBY

OBJECT\_NAME(i.object\_id)

-----------------------------------------------

SELECT

t.NAME AS TableName,

p.rowsAS RowCounts,

SUM(a.total\_pages)\* 8 / 1024 / 1024 AS TotalSpaceGB,

SUM(a.used\_pages)\* 8 / 1024 / 1024 AS UsedSpaceGB,

(SUM(a.total\_pages)-SUM(a.used\_pages))\* 8 / 1024 / 1024 AS UnusedSpaceGB

FROM

sys.tables t

INNERJOIN

sys.indexes i ON t.OBJECT\_ID= i.object\_id

INNERJOIN

sys.partitions p ON i.object\_id= p.OBJECT\_IDAND i.index\_id = p.index\_id

INNERJOIN

sys.allocation\_units a ON p.partition\_id= a.container\_id

LEFTOUTERJOIN

sys.schemas s ON t.schema\_id= s.schema\_id

WHERE

--t.NAME = 'YourTable'

t.is\_ms\_shipped = 0

AND i.OBJECT\_ID> 255

GROUPBY

t.Name, s.Name, p.Rows

ORDERBY

UsedSpaceGB DESC, t.Name

# T-sql query to find the biggest table in a database with a clustered index

SELECT

'['+(OBJECT\_SCHEMA\_NAME(tables.object\_id,db\_id())

+'].['+tables.NAME+']')AS TableName,

(sum(allocation\_units.total\_pages)\* 8)/ 1024 as TotalSpaceMB

FROM

sys.tablestables

INNERJOIN

sys.indexesindexesONtables.OBJECT\_ID=indexes.object\_id

INNERJOIN

sys.partitionspartitionsONindexes.object\_id=partitions.OBJECT\_ID

ANDindexes.index\_id =partitions.index\_id

INNERJOIN

sys.allocation\_unitsallocation\_unitsONpartitions.partition\_id=allocation\_units.container\_id

WHERE

indexes.index\_id = 1

GROUPBY

tables.object\_id,tables.NAME,indexes.object\_id,indexes.index\_id,indexes.name

ORDERBY

TotalSpaceMB desc

**Check the database backup details**

SELECTTOP 100

s.database\_name,

m.physical\_device\_name,

CAST(DATEDIFF(second, s.backup\_start\_date,

s.backup\_finish\_date)ASVARCHAR(100))+' '+'Seconds' TimeTaken,

s.backup\_start\_date,

CASE s.[type]

WHEN'D'THEN'Full'

WHEN'I'THEN'Differential'

WHEN'L'THEN'Transaction Log'

ENDAS BackupType,

s.server\_name,

s.recovery\_model

FROM msdb.dbo.backupset s

INNERJOIN msdb.dbo.backupmediafamily m ON s.media\_set\_id = m.media\_set\_id

--WHERE s.database\_name = DB\_NAME() -- Remove this line for all the database

ORDERBY backup\_start\_date DESC, backup\_finish\_date

GO

**Creating the Filegroup for table assigned to ndf files**

USEmaster

GO

ALTERDATABASECOMMON\_test

ADDFILEGROUPTest1FG1

GO

ALTERDATABASECOMMON\_test

ADDFILE

(NAME=COMMON\_test02,

FILENAME='C:\DB\_Files\COMMON\_test02.ndf',

SIZE= 100MB,

FILEGROWTH=5GB)

TOFILEGROUPTest1FG1

CREATEUNIQUECLUSTEREDINDEXPK\_EF\_CASE\_TYPE\_MAIN<Indexname>

ONdbo.EF\_CASE\_TYPE\_MAIN(CASE\_TYPE\_CODE<Column>)

WITHDROP\_EXISTING

ON[Test1FG1<filegroup>]

CREATETABLEdbo.employee

(emp\_idint,

emp\_fnamevarchar(10),

emp\_lnamevarchar(10))on[Test1FG1]

GO

**Check the name of the node on which the clustered SQL Server instance is running on**

SELECTSERVERPROPERTY('ComputerNamePhysicalNetBIOS')AS [CurrentNodeName]

SELECT\*FROMsys.dm\_os\_cluster\_nodes

**Permission for stored Procedure**

REVOKE EXECUTE on AA\_SSO\_Deactivation to<user>

GRANT EXECUTE on AA\_SSO\_Deactivation to<user>

GRANT EXECUTET O<user>

GRANT EXECUTE on DATABASE::STAGING\_MIGRATION TO Test

GRANT VIEW DEFINITION ON [dbo].[usp\_name] TO [user]

# [insert a NOT NULL column to an existing table](http://stackoverflow.com/questions/3492947/insert-a-not-null-column-to-an-existing-table)

ALTER TABLE TBL\_BAIL ADD SECURITY\_NO [nvarchar](200) NOT NULL DEFAULT'0'

ALTER TABLE TBL\_BAIL\_DOCUMENT DROPCOLUMN SECURITY\_NO

**Permission for Tables**

GRANT SELECT,INSERT,UPDATE,DELETE ON dbo.TBL\_EMAIL\_OUTBOX TO EAIUser;

REVOKE SELECT,INSERT,UPDATE,DELETE ON dbo.TBL\_EMAIL\_OUTBOX TO EAIUser;

GRANT CREATE TABLE TO Test;

GRANT ALTER ON SCHEMA::dbo TO Test;

**1.** db\_datareader will allow SELECT access to all tables

**2.** db\_datarwriter will allow INSERT, UPDATE, and DELETE access to all tables

**Display View columns**

Select a.name View\_name,b.name column\_name

Fromsys.all\_objects a,sys.all\_columns b

where a.object\_id=b.object\_id

and a.type='V'

and a.name like'%ALL\_WA\_EMP\_DATA%'

**Access to already existing user**

USE YourDB  
GO  
EXEC sp\_change\_users\_login 'update\_one', 'username', 'username'  
GO

EXECsp\_change\_users\_login'AUTO\_FIX','user'

**How to Identify the Default Database for a Login:**  
SELECT name AS LoginName, default\_database\_name   
FROM sys.server\_principals   
WHERE type\_desc = 'SQL\_LOGIN';  
  
Resolution:  
Updated the login’s default database to master using:  
  
ALTER LOGIN [YourLogin] WITH DEFAULT\_DATABASE = [master];

**Access to one view to user**

GRANT select on dbo.ALL\_WA\_EMP\_DATA TO <user>

REVOKE select on dbo.ALL\_WA\_EMP\_DATA TO <user>

DENY select on dbo.ALL\_WA\_EMP\_DATA TO <user>

**Access to one table to user**

GRANT SELECT ON [dbo].[JobStatsHistory] TO [Madhu]

**Access to database to user**

GRANT SELECT,INSERT,UPDATE,DELETE ON DATABASE::CMS TO Test

REVOKE SELECT,INSERT,UPDATE,DELETE ON DATABASE::CMS TO Test

DENY SELECT,INSERT,UPDATE,DELETE ON DATABASE::CMS TO Test

GRANT SELECT ON DATABASE::CMS TO Test

REVOKE SELECT ON DATABASE::CMS TO Test

DENY SELECT ON DATABASE::CMS TO Test

**Grant permission to create table**

USE BKM\_STG

GRANT ALTER ON Schema:: dbo TO omesti\_maricar

GRANT CREATE TABLE TO omesti\_maricar

GO

**Grant permission to schema**

GRANT ALTER,EXECUTE,SELECT,DELETE,INSERT,UPDATE,REFERENCES,VIEWDEFINITION,CREATESEQUENCE

ONSCHEMA::dboTO [user]

Go

GRANTALTERONSCHEMA::dbo TOuser;

GO

REVOKE EXECUTE,SELECT,DELETE,INSERT,UPDATE,REFERENCES,VIEWDEFINITION,CREATESEQUENCE

ON SCHEMA::dbo FROM user CASCADE;

GRANT CREATE PROCEDURETO BKM

GRANTSELECT, INSERT, UPDATE, DELETEONSCHEMA::dbo TO[user]

GRANT SELECT ON schema::dboTO testUser

GO

GRANT SELECT ON SCHEMA::dbo TO ReadOnlyRole;

GRANT ALTER ON SCHEMA::dbo TO [user]  
GRANT EXECUTE ON SCHEMA::dbo TO [user]  
GRANT DELETE ON SCHEMA::dbo TO [user]  
  
GRANT CREATE VIEW TO [user]  
GRANT CREATE FUNCTION TO [user]

GRANT SHOWPLAN TO Test

GRANT INSERT ON [log].[DatabaseChangeLog] TO [user]-- used by DDL trigger

**List of all stored procedure**

SELECT[schema]=OBJECT\_SCHEMA\_NAME([object\_id]),

name

FROMsys.procedures;

**List all users role and permissions**

select princ.name

, princ.type\_desc

, perm.permission\_name

, perm.state\_desc

, perm.class\_desc

,object\_name(perm.major\_id)

fromsys.database\_principals princ

leftjoin

sys.database\_permissions perm

on perm.grantee\_principal\_id = princ.principal\_id

**How to check the orphan user**

First, make sure that this is the problem. This will lists the orphaned users:

EXEC sp\_change\_users\_login 'Report'

If you already have a login id and password for this user, fix it by doing:

EXEC sp\_change\_users\_login 'Auto\_Fix', '*user*'

Update\_One’ links the specified user in the current database to login. login must already exist.

EXEC sp\_change\_users\_login 'update\_one', 'ColdFusion', 'ColdFusion'

If you want to create a new login id and password for this user, fix it by doing:

EXEC sp\_change\_users\_login 'Auto\_Fix', '*user*', '*login*', '*password*'

EXEC sp\_change\_users\_login'Report'

--------------------------------------------------------

DECLARE @username varchar(25)

DECLARE fixusers CURSOR

FOR

SELECT UserName = name FROMsysusers

WHERE issqluser = 1 and(sid is not null and sid<> 0x0)

And suser\_sname(sid)isnull

ORDER BY name

OPEN fixusers

FETCH NEXT FROM fixusers

INTO @username

WHILE@@FETCH\_STATUS= 0

BEGIN

EXEC sp\_change\_users\_login'update\_one', @username, @username

FETCHNEXTFROM fixusers

INTO @username

END

CLOSE fixusers

DEALLOCATE fixusers

**Display all sql agent Jobs**

SELECT

job.job\_id,

notify\_level\_email,

name,

enabled,

description,

step\_name,

command,

server,

database\_name

FROM

msdb.dbo.sysjobs job

INNERJOIN

msdb.dbo.sysjobsteps steps

ON

job.job\_id = steps.job\_id

**Check the recent expensive queries SPID**

SELECT\*FROMsys.dm\_exec\_query\_memory\_grants

Select spid,hostname,program\_name,nt\_username,loginame,cmd, t.textfromsys.sysprocesses

CROSSAPPLY(selecttextfromsys.dm\_exec\_sql\_text(sql\_handle))t

**Recent running queries with date stamp**

SELECT st.text,

qp.query\_plan,

qs.\*

FROM (

SELECTTOP 50 \*

FROMsys.dm\_exec\_query\_stats

ORDERBY total\_worker\_time DESC

)AS qs

CROSSAPPLYsys.dm\_exec\_sql\_text(qs.sql\_handle)AS st

CROSSAPPLYsys.dm\_exec\_query\_plan(qs.plan\_handle)AS qp

WHERE qs.max\_worker\_time > 300

OR qs.max\_elapsed\_time > 300

**===============================================================**

SELECTtop 10

DB\_NAME(st.dbid)ASDatabaseName

,OBJECT\_SCHEMA\_NAME(st.objectid,dbid)ASSchemaName

,cp.objtypeASObjectType

,OBJECT\_NAME(st.objectid,dbid)ASObjects

,MAX(cp.usecounts)ASTotal\_Execution\_count

,SUM(qs.total\_worker\_time)ASTotal\_CPU\_Time

,SUM(qs.total\_worker\_time)/(max(cp.usecounts)\* 1.0)ASAvg\_CPU\_Time

FROMsys.dm\_exec\_cached\_planscp

INNERJOINsys.dm\_exec\_query\_statsqs

ONcp.plan\_handle=qs.plan\_handle

CROSSAPPLYsys.dm\_exec\_sql\_text(cp.plan\_handle)st

WHEREDB\_NAME(st.dbid)ISNOTNULL

GROUPBYDB\_NAME(st.dbid),OBJECT\_SCHEMA\_NAME(objectid,st.dbid),cp.objtype,OBJECT\_NAME(objectid,st.dbid)

ORDERBYsum(qs.total\_worker\_time)desc

===============================================================

SELECTTOP 10 SUBSTRING(qt.TEXT,(qs.statement\_start\_offset/2)+1,

((CASE qs.statement\_end\_offset

WHEN-1 THENDATALENGTH(qt.TEXT)

ELSE qs.statement\_end\_offset

END- qs.statement\_start\_offset)/2)+1),

qs.execution\_count,

qs.total\_logical\_reads, qs.last\_logical\_reads,

qs.total\_logical\_writes, qs.last\_logical\_writes,

qs.total\_worker\_time,

qs.last\_worker\_time,

qs.total\_elapsed\_time/1000000 total\_elapsed\_time\_in\_S,

qs.last\_elapsed\_time/1000000 last\_elapsed\_time\_in\_S,

qs.last\_execution\_time,

qp.query\_plan

FROMsys.dm\_exec\_query\_stats qs

CROSSAPPLYsys.dm\_exec\_sql\_text(qs.sql\_handle) qt

CROSSAPPLYsys.dm\_exec\_query\_plan(qs.plan\_handle) qp

--ORDER BY qs.total\_logical\_reads DESC -- logical reads

--ORDER BY qs.total\_logical\_writes DESC -- logical writes

ORDERBY qs.total\_worker\_time DESC-- CPU time

### Query to find Top 10 Adhoc Queries

SELECT q.[text],

(highest\_cpu\_queries.total\_worker\_time/highest\_cpu\_queries.execution\_count)/ 1000000.0 AS AverageCPU,

highest\_cpu\_queries.execution\_count as distinctCalls, highest\_cpu\_queries.total\_worker\_time,

highest\_cpu\_queries.last\_execution\_time,

highest\_cpu\_queries.last\_physical\_reads, highest\_cpu\_queries.min\_physical\_reads,

highest\_cpu\_queries.sql\_handle

FROM (SELECTTOP 10 qs.sql\_handle, qs.total\_worker\_time, qs.last\_execution\_time,

qs.plan\_generation\_num, qs.execution\_count, qs.last\_physical\_reads,

qs.min\_physical\_reads FROMsys.dm\_exec\_query\_stats qs

ORDERBY qs.total\_worker\_time DESC)AS highest\_cpu\_queries

CROSSAPPLYsys.dm\_exec\_sql\_text(sql\_handle)AS q

ORDERBY AverageCPU DESC

============================================

WITH cte AS

(

SELECT stat.[sql\_handle],

stat.statement\_start\_offset,

stat.statement\_end\_offset,

COUNT(\*)AS [NumExecutionPlans],

SUM(stat.execution\_count)AS [TotalExecutions],

((SUM(stat.total\_logical\_reads)\* 1.0)/SUM(stat.execution\_count))AS [AvgLogicalReads],

((SUM(stat.total\_worker\_time)\* 1.0)/SUM(stat.execution\_count))AS [AvgCPU]

FROMsys.dm\_exec\_query\_stats stat

GROUPBY stat.[sql\_handle], stat.statement\_start\_offset, stat.statement\_end\_offset

)

SELECTCONVERT(DECIMAL(15, 5), cte.AvgCPU)AS [AvgCPU],

CONVERT(DECIMAL(15, 5), cte.AvgLogicalReads)AS [AvgLogicalReads],

cte.NumExecutionPlans,

cte.TotalExecutions,

DB\_NAME(txt.[dbid])AS [DatabaseName],

OBJECT\_NAME(txt.objectid, txt.[dbid])AS [ObjectName],

SUBSTRING(txt.[text],(cte.statement\_start\_offset / 2)+ 1,

(

(CASE cte.statement\_end\_offset

WHEN-1 THENDATALENGTH(txt.[text])

ELSE cte.statement\_end\_offset

END- cte.statement\_start\_offset)/ 2

)+ 1

)

FROM cte

CROSSAPPLYsys.dm\_exec\_sql\_text(cte.[sql\_handle]) txt

ORDERBY cte.AvgCPU DESC;

**Check the stored procedure query plan**

SELECT

OBJECT\_NAME(st.objectid,st.dbid)ASObjectName,

cp.usecountsASExecutionCount,

st.TEXTASQueryText,

qp.query\_planASQueryPlan

FROM

sys.dm\_exec\_cached\_plansAScp

CROSSAPPLYsys.dm\_exec\_query\_plan(cp.plan\_handle)ASqp

CROSSAPPLYsys.dm\_exec\_sql\_text(cp.plan\_handle)ASst

WHERE

cp.objtype='Proc'

ANDOBJECT\_NAME(st.objectid,st.dbid)='YourStoredProcedure';

**Show the database connection pools**

select loginame , nt\_username,count(\*)'Connections count'

frommaster..sysprocesses

where spid > 50 and spid !=@@spid

groupby loginame , nt\_username

orderbycount(\*)desc

-----------------------------------

Select\*fromSys.sysprocesses

Where spid>50

Anddbid=db\_id('COMMON')

---------------------------------------

SELECT

DB\_NAME(dbid)asDBName,

COUNT(dbid)asNumberOfConnections,

loginameasLoginName

FROM

sys.sysprocesses

WHERE

dbid> 0

GROUPBY

dbid,loginame

**Creating linked server**

EXECsp\_addlinkedserver

@server='FRDEVDOTNETDEV\SHUN',--Provide Linked Server Name

@srvproduct='',

@provider='sqlncli',

@datasrc='tcp:ecourts-app-01.southeastasia.cloudapp.azure.com.database.windows.net,1433',--provide Azure SQl Server name

@location='',

@provstr='',

@catalog=''--Provide azure database name

execsp\_addlinkedsrvlogin'FRDEVDOTNETDEV\SHUN','FALSE',NULL,'sa','p@ssw0rd';

execsp\_addlinkedsrvlogin'192.168.1.42','FALSE',NULL,'sa','p@$$w0rd';

**Creating new Unidentifier**

selectNEWID()

selectCAST(HASHBYTES('MD5',ISNULL('A',''))ASUNIQUEIDENTIFIER)

**Restore the database**

ALTERDATABASE EFSDb

SETSINGLE\_USERWITH

ROLLBACKIMMEDIATE

----Restore Database

RESTOREDATABASE EFSDb

FROMDISK='N:\UAT\_BACKUPS\Full\EFSDb\_30112016\_0946.bak'

WITHMOVE'EFSDb'TO'E:\MSSQL12.SIT\MSSQL\Data\EFSDb.mdf',

MOVE'EFSDb\_log'TO'L:\MSSQL12.SIT\MSSQL\Data\EFSDb\_log.ldf',

REPLACE

ALTERDATABASE EFSDb SETMULTI\_USER

GO

--------------------------------

ALTERDATABASE EFSDb SETSINGLE\_USERWITHROLLBACKIMMEDIATE

GO

RESTOREDATABASE EFSDb FROMDISK='N:\UAT\_BACKUPS\Full\EFSDb\EFSDb\_23012017\_1737.BAK'WITHREPLACE

GO

ALTERDATABASE EFSDb SETMULTI\_USER;

GO

**Merge the NDF file in one MDF files**

DBCC SHRINKFILE('logical\_ndf\_file\_name', EMPTYFILE);

ALTERDATABASE database\_name REMOVE FILE logical\_ndf\_file\_name;

**Merge the LDF file in one Log LDF files**

Take the database into simple mode

DBCCSHRINKFILE('CMS\_02',EMPTYFILE)

USEmaster

GO

ALTERDATABASECMSREMOVEFILECMS\_02

GO

<https://stackoverflow.com/questions/41186473/is-it-possible-to-merge-ndf-files-and-mdf-file-into-one-single-mdf-file>

<https://www.mssqltips.com/sqlservertip/4419/renaming-physical-database-file-names-for-a-sql-server-database/>

SETIDENTITY\_INSERT database.dbo.tbl ON

**Permission to user on selecting tables**

GRANTVIEWSERVERSTATETOomesti\_kah\_Ching

REVOKEVIEWSERVERSTATETOomesti\_kah\_Ching

DENYVIEWSERVERSTATETOomesti\_kah\_Ching

**Create and alter existing schema**

IF (NOTEXISTS(SELECT\*FROMsys.schemasWHEREname='exe'))

BEGIN

EXEC ('CREATE SCHEMA [exe] AUTHORIZATION [dbo]')

END

ALTERSCHEMAexe

TRANSFERdbo.EF\_APPEAL\_COURT\_UNIT\_MAIN

**Trace the sql log file**

select\*fromfn\_trace\_getinfo(NULL)

where property=2

and traceid = 1

select\*fromfn\_trace\_gettable('D:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\Log\log\_19.trc',-1)where ClientProcessID =10936

**View the Audit file**

select\*from fn\_get\_audit\_file('T:\Trace\_file\Audit-20161227-151559\_2B3199B2-952F-4C5A-98D3-8A70849811E0\_0\_131325113569200000.sqlaudit',default,default)

SELECT\*FROM(SELECTCOALESCE(OBJECT\_NAME(s2.objectid),'Ad-Hoc')AS ProcName,

execution\_count,s2.objectid,

(SELECTTOP 1 SUBSTRING(s2.TEXT,statement\_start\_offset / 2+1 ,

((CASEWHEN statement\_end\_offset =-1

THEN (LEN(CONVERT(NVARCHAR(MAX),s2.TEXT))\* 2)

ELSE statement\_end\_offset END)- statement\_start\_offset)/ 2+1))AS sql\_statement,

last\_execution\_time

FROMsys.dm\_exec\_query\_statsAS s1

CROSSAPPLYsys.dm\_exec\_sql\_text(sql\_handle)AS s2 ) x

WHERE sql\_statement NOTlike'SELECT \* FROM(SELECT %'

--and OBJECTPROPERTYEX(x.objectid,'IsProcedure') = 1

ORDERBY last\_execution\_time DESC

**Auto schedule any script in Job agent**

**Connection Pool**

declare @sql varchar(8000)

select @sql = 'SQLCMD -S POJPRDEFSDB01 -d master -E -Q "select loginame , nt\_username,count(\*) AS [Connections count] from master..sysprocesses where spid > 50 and spid !=@@spid group by loginame , nt\_username order by count(\*)desc" -b >> N:\Full\myoutput.txt'

exec xp\_cmdshell 'echo %date% %time% >> N:\Full\myoutput.txt'

exec master..xp\_cmdshell @sql, no\_output

**SP\_WHO2**

declare @sql varchar(8000)

select @sql = 'SQLCMD -S POJPRDEFSDB01 -d master -E -Q "sp\_who2" -b |findstr /v /c:"---" >> c:\myoutput.txt '

exec xp\_cmdshell 'echo %date% %time% >> c:\myoutput1.txt'

exec master..xp\_cmdshell @sql, no\_output

**Change the servername to computer name**

SELECTSERVERPROPERTY(N'servername')

SELECT@@SERVERNAME

DECLARE @actualname NVARCHAR(100)

DECLARE @currentname NVARCHAR(100)

SELECT @actualname = CONVERT(NVARCHAR(100), SERVERPROPERTY(N'servername'))

SELECT @currentname = @@SERVERNAME

EXEC sp\_dropserver @currentname

EXEC sp\_addserver @actualname, local

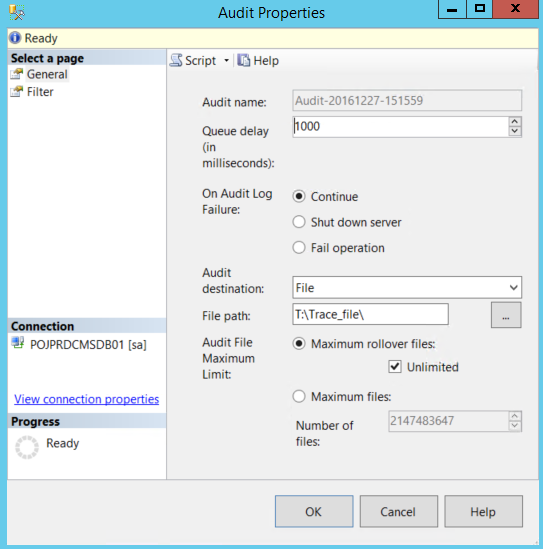
EXECsp\_DROPSERVER 'oldservername'

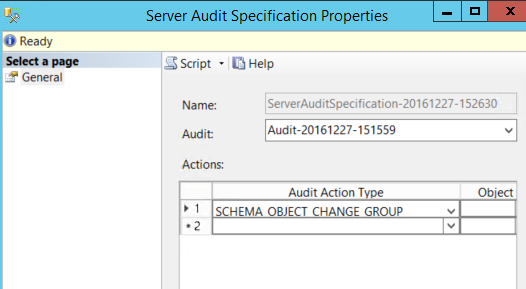
EXECsp\_ADDSERVER 'newservername', 'local'

**Create and insert table data**

Select\*intoEmployee\_tempfromTest.dbo.Employee

**Creating the Audit log**





**Find the table collationinformation**

execsp\_help'[dbo].[E\_Filing]'

**List of all Tables,Views, stored\_procedure**

SELECT\*FROMinformation\_schema.tables

SELECT\*FROMinformation\_schema.VIEWS

select\*fromINFORMATION\_SCHEMA.ROUTINES

**List of table all columns,data type and length**

SELECT

c.name'Column Name',

t.name,

t.name+

CASEWHEN t.nameIN('char','varchar','nchar','nvarchar')THEN'('+

CASEWHEN c.max\_length=-1 THEN'MAX'

ELSECONVERT(VARCHAR(4),

CASEWHEN t.nameIN('nchar','nvarchar')

THEN c.max\_length/2 ELSE c.max\_length END)

END+')'

WHEN t.nameIN('decimal','numeric')

THEN'('+CONVERT(VARCHAR(4),c.precision)+','

+CONVERT(VARCHAR(4),c.Scale)+')'

ELSE''END

as "DDL name",

c.max\_length 'Max Length in Bytes',

c.precision,

c.scale ,

c.is\_nullable,

ISNULL(i.is\_primary\_key, 0)'Primary Key'

FROM

sys.columns c

INNERJOIN

sys.types t ON c.user\_type\_id = t.user\_type\_id

LEFTOUTERJOIN

sys.index\_columns ic ON ic.object\_id= c.object\_idAND ic.column\_id = c.column\_id

LEFTOUTERJOIN

sys.indexes i ON ic.object\_id= i.object\_idAND ic.index\_id = i.index\_id

WHERE

c.object\_id=OBJECT\_ID('Dimension\_FAProduct')

**Find the table columns created data and information**

SELECT

crdate as thisFieldIsNotTheCreationOfTheField1 ,

refdate as thisFieldIsNotTheCreationOfTheField2 ,

o.name AS[TableName],

o.type,

c.name AS[ColName],

s.name AS[ColType],

c.prec,

c.scale,

c.isnullable

FROM

dbo.sysobjects AS o

INNERJOIN

dbo.syscolumns AS c ON c.id = o.id

INNERJOIN

dbo.systypes AS s ON c.xtype = s.xtype

WHERE

o.type ='U'and o.name='TableNa'

ORDERBY

crdate

**Find the index created date details**

select

crdate,i.name,object\_name(o.id)

from

sysindexesi

join

sysobjectsoONo.id=i.id

where

i.namelike'%PK%'or

i.namelike'%IDX%'or

i.namelike'%IX%'

**Usage of index**

SELECT'['+Sch.name+'].['+Tab.[name]+']'ASTableName,

Ind.type\_desc,

Ind.[name]ASIndexName,

SUBSTRING((SELECT', '+AC.name

FROMsys.[tables]AST

INNERJOINsys.[indexes]IONT.[object\_id]=I.[object\_id]

INNERJOINsys.[index\_columns]ICONI.[object\_id]=IC.[object\_id]

ANDI.[index\_id]=IC.[index\_id]

INNERJOINsys.[all\_columns]ACONT.[object\_id]=AC.[object\_id]

ANDIC.[column\_id]=AC.[column\_id]

WHEREInd.[object\_id]=I.[object\_id]

ANDInd.index\_id=I.index\_id

ANDIC.is\_included\_column= 0

ORDERBYIC.key\_ordinal

FOR

XMLPATH('')), 2, 8000)ASKeyCols,

SUBSTRING((SELECT', '+AC.name

FROMsys.[tables]AST

INNERJOINsys.[indexes]IONT.[object\_id]=I.[object\_id]

INNERJOINsys.[index\_columns]ICONI.[object\_id]=IC.[object\_id]

ANDI.[index\_id]=IC.[index\_id]

INNERJOINsys.[all\_columns]ACONT.[object\_id]=AC.[object\_id]

ANDIC.[column\_id]=AC.[column\_id]

WHEREInd.[object\_id]=I.[object\_id]

ANDInd.index\_id=I.index\_id

ANDIC.is\_included\_column= 1

ORDERBYIC.key\_ordinal

FOR

XMLPATH('')), 2, 8000)ASIncludeCols,

usg\_stats.user\_seeksASUserSeek,

usg\_stats.user\_scansASUserScans,

usg\_stats.user\_lookupsASUserLookups,

usg\_stats.user\_updatesASUserUpdates

FROMsys.[indexes]ASInd

INNERJOINsys.[tables]ASTabONTab.[object\_id]=Ind.[object\_id]

INNERJOINsys.[schemas]ASSchONSch.[schema\_id]=Tab.[schema\_id]

LEFTOUTERJOINsys.dm\_db\_index\_usage\_statsASusg\_statsONInd.index\_id=usg\_stats.index\_id

ANDInd.[OBJECT\_ID]=usg\_stats.[OBJECT\_ID]andusg\_stats.database\_id=DB\_ID()

WHEREInd.type\_desc<>'HEAP'

--AND Tab.name = 'YourTableNameHere' -- uncomment to get single table indexes detail

ORDERBYTableName

### Limit Concurrent Logins by database and/or user ID

USE master;

GO

CREATE LOGIN login\_test WITH PASSWORD = '3KHJ6dhx(0xVYsdf' MUST\_CHANGE,

CHECK\_EXPIRATION = ON;

GO

GRANT VIEW SERVER STATE TO login\_test;

GO

CREATE TRIGGER connection\_limit\_trigger

ON ALL SERVER WITH EXECUTE AS 'login\_test'

FOR LOGON

AS

BEGIN

IF ORIGINAL\_LOGIN()= 'login\_test' AND

(SELECT COUNT(\*) FROM sys.dm\_exec\_sessions

WHERE is\_user\_process = 1 AND

original\_login\_name = 'login\_test') > 3

ROLLBACK;

END;

**SQL Server Buffer Manager metrics and memory counters**

SELECTobject\_name, counter\_name, cntr\_value

FROMsys.dm\_os\_performance\_counters

WHERE [object\_name] LIKE'%Buffer Manager%'

AND [counter\_name] ='Page reads/sec'

SELECTobject\_name, counter\_name, cntr\_value

FROMsys.dm\_os\_performance\_counters

WHERE [object\_name] LIKE'%Buffer Manager%'

AND [counter\_name] ='Page writes/sec'

**Changing Logshipping from NO RECOVERY to STANDBY Mode**

USE[Master]

GO

executesp\_change\_log\_shipping\_secondary\_database

@secondary\_database='DBName'

,@restore\_mode= 1

,@disconnect\_users= 0 -- or 1

--Depends on your choice. Disconnect is safer but more annoying for users

-------------------------------------------------------------------

RESTOREDATABASE[MyDatabase]

WITHSTANDBY=N'D:\MSSQL\Backup\MyDatabase\_RollbackUndoFile.bak'

restoredatabasedb\_namewithrecovery

**stored procedure was last executed**

SELECTa.execution\_count,

OBJECT\_NAME(objectid)Name,

query\_text=SUBSTRING(

b.text,

a.statement\_start\_offset/2,

(CASEWHENa.statement\_end\_offset=-1

THENlen(convert(nvarchar(max),b.text))\* 2

ELSEa.statement\_end\_offset

END-a.statement\_start\_offset)/2

),

b.dbid,

dbname=db\_name(b.dbid),

b.objectid,

a.creation\_time,

a.last\_execution\_time,

a.\*

FROMsys.dm\_exec\_query\_statsa

CROSSAPPLYsys.dm\_exec\_sql\_text(a.sql\_handle)asb

WHEREOBJECT\_NAME(objectid)='USP\_SELT\_DOCUMENT\_PROCESSING'

ORDERBYa.last\_execution\_timeDESC

**Find the session transcation**

select\*fromsys.dm\_exec\_sessions

**Check the database restoration status**

SELECTsession\_idasSPID,command,a.textASQuery,start\_time,percent\_complete,dateadd(second,estimated\_completion\_time/1000,getdate())asestimated\_completion\_time

FROMsys.dm\_exec\_requestsrCROSSAPPLYsys.dm\_exec\_sql\_text(r.sql\_handle)a

WHEREr.commandin('BACKUP DATABASE','RESTORE DATABASE')

SELECT command, percent\_complete,

'elapsed'= total\_elapsed\_time / 60000.0,

'remaining'= estimated\_completion\_time / 60000.0

FROMsys.dm\_exec\_requests

WHERE command like'BACKUP%'

or command like'RESTORE%'

**MS SQL server health checkup**

<http://ajaydwivedi.com/2018/03/sp_healthcheck-get-sql-server-health-details/>

<https://github.com/imajaydwivedi/SQLDBA-SSMS-Solution/tree/master/sp_HealthCheck>

**Check the job history details**

select \* from msdb.dbo.sysjobhistory;

**To know how much physical memory SQL Server**

select

(physical\_memory\_in\_use\_kb/1024)Phy\_Memory\_usedby\_Sqlserver\_MB,

(locked\_page\_allocations\_kb/1024 )Locked\_pages\_used\_Sqlserver\_MB,

(virtual\_address\_space\_committed\_kb/1024 )Total\_Memory\_UsedBySQLServer\_MB,

process\_physical\_memory\_low,

process\_virtual\_memory\_low

from sys. dm\_os\_process\_memory

Restart the DB from cmd prompt

1. Stop the instance of SQL Server if it is started.
2. Start the instance of SQL Server in master-only recovery mode by entering one of the following commands at the command prompt.

For the default (MSSQLSERVER) instance, run the following command.

NET START MSSQLSERVER /f /T3608

For a named instance, run the following command.

NET START MSSQL$instancename /f /T3608

https://github.com/PNNL-Comp-Mass-Spec/DBSchema\_MTS/blob/master/dba/rpt\_HealthReport.sql