SQL Kommando Håndbog

(Mikael Veistrup-Vetlov)

Indhold

[Standard MSSQL DBA procedurer 6](#_Toc93671597)

[Installation af SQL Server instans 6](#_Toc93671598)

[Uninstall af SQL Server instans 6](#_Toc93671599)

[Start og stop af SQL Instans 6](#_Toc93671600)

[Ret SQL opstarts parametre / Trace flags 7](#_Toc93671601)

[Oprettelse af MSSQL DBA login til SQL, hvis den mangler 8](#_Toc93671602)

[Opret Ny SQL Database 8](#_Toc93671603)

[Slet database der er offline 8](#_Toc93671604)

[DB\_Owner & anbefalinger 8](#_Toc93671605)

[-- Change job ownership of single job to sa 9](#_Toc93671606)

[--Vis alle Aktive kørende Jobs I SQL agent. 10](#_Toc93671607)

[How to stop a runaway SSIS package 12](#_Toc93671608)

[SQL Locks & Deadlock 13](#_Toc93671609)

[Procedure til søgning af deadlocks via T-Sql 14](#_Toc93671610)

[List antal Deadlock forekomster siden SQL engine start. 14](#_Toc93671611)

[Via Extended events 15](#_Toc93671612)

[Begræns Memory Used 15](#_Toc93671613)

[Find SQL Recovery Model, Last Backup & Backup type 16](#_Toc93671614)

[Oversigt over Extended events. 16](#_Toc93671615)

[Audit Logning fra Extended events. 17](#_Toc93671616)

[Gem Check\_MK databaseliste I tabel 19](#_Toc93671617)

[Hent nyeste Check\_MK databaseliste fra tabel 19](#_Toc93671618)

[Denne del danner tekst strengen til Check\_mk (nedenstående er med et dårlige eksempel) 20](#_Toc93671619)

[Hent SQL tekst strengen til Check\_mk 20](#_Toc93671620)

[Check om Kolonne findes I Tabel. 20](#_Toc93671621)

[SQL Adgang 20](#_Toc93671622)

[Standard opsætning med Læs, Skriv, dbo grupper. 20](#_Toc93671623)

[SQL Login – adgang til SQL server Instans 20](#_Toc93671624)

[DB User Tilknyt Login til DataBase (kræver Login) 20](#_Toc93671625)

[DB Privilegier til data på DataBase (kræver Login & bruger oprettet på db) 20](#_Toc93671626)

[DB Privilegier på view (kræver Login & bruger oprettet på db) 20](#_Toc93671627)

[Opret SQL Sysadmin login Emergency 21](#_Toc93671628)

[Styring af SQL agent jobs & roller 21](#_Toc93671629)

[Se hvem der har SQLagentUserRole roller mm. 22](#_Toc93671630)

[Activity Monitor tool - Statistics 22](#_Toc93671631)

[Database dm Sys.dm\_db Database Statistikker 22](#_Toc93671632)

[Se hvem der har VIEW SERVER STATE privilegier 23](#_Toc93671633)

[Login Errors (18456) 23](#_Toc93671634)

[Find Server rolle medlemmer 24](#_Toc93671635)

[find Instance Logins with default DB other than master & msdb 24](#_Toc93671636)

[find Instance Logins with more than user access. Shows Sysadmins 25](#_Toc93671637)

[find DB users with more than user access. Shows db\_owner 26](#_Toc93671638)

[Find DB users with only user access 27](#_Toc93671639)

[Find db users who are connected to Instance Login via sid 27](#_Toc93671640)

[Find orphaned db users with broken links 27](#_Toc93671641)

[Slet orphaned db user after migration 28](#_Toc93671642)

[Tilknyt orphaned db user til instance login. 28](#_Toc93671643)

[Change all databases to Recovery Full 29](#_Toc93671644)

[Change all tables in all databases to ? 29](#_Toc93671645)

[Allow & Deny SQL Users & Logins 30](#_Toc93671646)

[Tillad execute på stored procedure: 30](#_Toc93671647)

[Escape Tokens til at overføre SQL info til Powershell Jobbene 30](#_Toc93671648)

[Klargør tom table, med same kolonner som den oprindelige. 31](#_Toc93671649)

[Tøm / Truncate /slet indhold i Database 31](#_Toc93671650)

[Kopier table til temp table 31](#_Toc93671651)

[List kolonne pr. Tabel pr. Database Søg CPR 31](#_Toc93671652)

[-- Hent alle linier hvor der er fundet ”cpr” pr. tabel eller kolonne 31](#_Toc93671653)

[-- Tæller forekomster af ordet ”cpr” i kolonnenavne & Tabelnavne pr. database 32](#_Toc93671654)

[-- viser alle tabeller pr. database 32](#_Toc93671655)

[-- viser antal tabeller pr. database 32](#_Toc93671656)

[-- Show column pr. Table pr. Schema in Database relations 32](#_Toc93671657)

[--Schema Owner 32](#_Toc93671658)

[Alter Schema owner to dbo 33](#_Toc93671659)

[Drop all normal users 33](#_Toc93671660)

[List alle db-users på instans vælg user via Temp table 34](#_Toc93671661)

[Stop Running Processes 34](#_Toc93671662)

[Show Connections to db 35](#_Toc93671663)

[Kill Connections to db 35](#_Toc93671664)

[Take Offline / Online / Detache DB 35](#_Toc93671665)

[CheckDB 36](#_Toc93671666)

[T-SQL læs default data location fra Registry 37](#_Toc93671667)

[T-SQL Skriv default data location til Registry 37](#_Toc93671668)

[Error log location 37](#_Toc93671669)

[File Size 37](#_Toc93671670)

[Check size / størrelse Tempdb: 37](#_Toc93671671)

[Oprydning i gamle Sletbare Tempdb filer 38](#_Toc93671672)

[Andre check af Tempdb filer. 38](#_Toc93671673)

[Move Tempdb to new disk. 39](#_Toc93671674)

[Shrink Tempdb in place 40](#_Toc93671675)

[Check Brug af TempDB når denne vokser meget. 40](#_Toc93671676)

[Move Database Files / Flyt DB filer 43](#_Toc93671677)

[SHRINK DATABASE / FILE 43](#_Toc93671678)

[Vis VLF (virtual log files) for en db. 46](#_Toc93671679)

[Ryd op I Aktive Connections til DB 47](#_Toc93671680)

[List aktive forbindelser: 47](#_Toc93671681)

[List active SQL kald 48](#_Toc93671682)

[Kan ryddes ved at tage offline, og derefter online igen, men husk ”with rollback immediate” 48](#_Toc93671683)

[Men endnu bedre begræns til sysadmins 48](#_Toc93671684)

[Opret en SQL linked Server 48](#_Toc93671685)

[For at linke en hel SQL server og forbinde via egen windows konto: 48](#_Toc93671686)

[Det kan være en fordel at oprette en Linked Server til en enkelt Database 49](#_Toc93671687)

[Så man opretter en Linked Server til en DB med egen windows konto: 49](#_Toc93671688)

[For at slette Linked Server Login: 50](#_Toc93671689)

[Og slette Linked Server 50](#_Toc93671690)

[Rapport over installeret SQL 50](#_Toc93671691)

[Tilføjelse af funktioner til Installeret SQL Server inkl. Tablediff.exe. 50](#_Toc93671692)

[Find Databaser & Disk fil navn / placering 50](#_Toc93671693)

[Database (suspect) eller (recovery Pending) 52](#_Toc93671694)

[Tag en genstridig Database offline 52](#_Toc93671695)

[--Kopi af Database fra prod til test 53](#_Toc93671696)

[The state an SQL database is in can be checked by running the following query: 53](#_Toc93671697)

[Moving SQL Server disk to new Hardware / Storage 53](#_Toc93671698)

[Oprettelse af Certifikat til Server, til brug af f.eks. HTTPS fra SSRS eller IIS 55](#_Toc93671699)

[Deploy SSIS package. 56](#_Toc93671700)

[Rollback SSIS package. 56](#_Toc93671701)

[Check\_MK SSAS Cube age via Powershell. 57](#_Toc93671702)

[SSAS Analysis Services Adm 58](#_Toc93671703)

[Multidimensional Modeling (Adventure Works Tutorial) 58](#_Toc93671704)

[Tabular modeling (1400 compatibility level) 58](#_Toc93671705)

[SSAS Kuber ældre end 2 dage ( CubeAge) 58](#_Toc93671706)

[SSAS Kube Proccessering fejl 59](#_Toc93671707)

[SSRS Reporting Services Adm 60](#_Toc93671708)

[Reporting Services Configuration Manager 60](#_Toc93671709)

[Microsoft R Services 60](#_Toc93671710)

[DMV 63](#_Toc93671711)

[Sys.dm\_io\_virtual\_file\_stats 63](#_Toc93671712)

[sys.dm\_tran\_locks 63](#_Toc93671713)

[sys.dm\_os\_waiting\_tasks. 63](#_Toc93671714)

[sys.dm\_os\_wait\_stats. 63](#_Toc93671715)

[sys.dm\_exec\_session\_wait\_stats. 63](#_Toc93671716)

[Check Backup jobs 63](#_Toc93671717)

[sys.dm\_exec\_requests 63](#_Toc93671718)

[sys.dm\_exec\_sql\_text 63](#_Toc93671719)

[sys.dm\_server\_services 63](#_Toc93671720)

[Master.sys.sql\_modules 63](#_Toc93671721)

[Monitor tempdb: 63](#_Toc93671722)

[sys.dm\_db\_file\_space\_usage 63](#_Toc93671723)

[sys.dm\_db\_session\_space\_usage & sys.dm\_db\_task\_space\_usage 63](#_Toc93671724)

[Faste Parametre 64](#_Toc93671725)

[Extended properties 65](#_Toc93671726)

# Standard MSSQL DBA procedurer

## Installation af SQL Server instans

Fra installations mediet:

.\Setup.exe /QS /ACTION=Install /FEATURES=$SQLFeat $SQLrAccpt $SQLrcab /INSTANCENAME=$SQLInst /SQLSVCACCOUNT=$SQLdomain\svd\_sqle$SQLsShortn$ /SQLSVCSTARTUPTYPE=Automatic /AGTSVCACCOUNT=$SQLdomain\svd\_sqla$SQLsShortn$ /AGTSVCSTARTUPTYPE=Automatic /BROWSERSVCSTARTUPTYPE=Automatic /SQLSYSADMINACCOUNTS=$SQLdomain"\"$SQLdbaGrp /TCPENABLED=1 /IACCEPTSQLSERVERLICENSETERMS=1 /UPDATEENABLED=True /SECURITYMODE=SQL /SAPWD="$SQLsapw" /SQLTEMPDBDIR="T:\System\MSSQL13.MSSQLSERVER\MSSQL\TempDB\" /SQLTEMPDBLOGDIR="T:\System\MSSQL13.MSSQLSERVER\MSSQL\TempLog\" /SQLTEMPDBFILESIZE=128 /SQLTEMPDBFILEGROWTH=64 /SQLTEMPDBLOGFILESIZE=64 /SQLUSERDBDIR="R:\System\MSSQL13.MSSQLSERVER\MSSQL\Data\" /SQLUSERDBLOGDIR="S:\System\MSSQL13.MSSQLSERVER\MSSQL\Log\" /SQLBACKUPDIR="U:\System\MSSQL13.MSSQLSERVER\MSSQL\Backup\" /SQLCOLLATION="Danish\_Norwegian\_CI\_AS"

$SQLdomain="sst.dk"

$SQLsName="s-mivesql21-t"

$SQLsShortn="mivesq1"

$SQLsa="svd\_sqlsa\_fmsql1p"

$SQLInst=""

$SQLInstFeat="r" # feature could be r or blank

$SQLsapw="^r&3UdoPwBbr"

$gMSA="x" # kan sættes til x hvis der bruges group Managed Service Accounts til de 2 næste.

.\Setup.exe /QS /ACTION=Install /FEATURES=$SQLFeat $SQLrAccpt $SQLrcab /INSTANCENAME=$SQLInst

## Uninstall af SQL Server instans

Fra installations mediet:

Setup.exe /QS /Action=Uninstall /FEATURES=$SQLFeat $SQLrAccpt $SQLrcab /INSTANCENAME=$SQLInst

.\Setup.exe /QS /Action=Uninstall /FEATURES=”SQL” /INSTANCENAME=”MGMT”

.\Setup.exe /QS /Action=Uninstall /FEATURES=”RS” /INSTANCENAME=”MSSQLSERVER”

(husk at Slette tempdb)

## Start og stop af SQL Instans

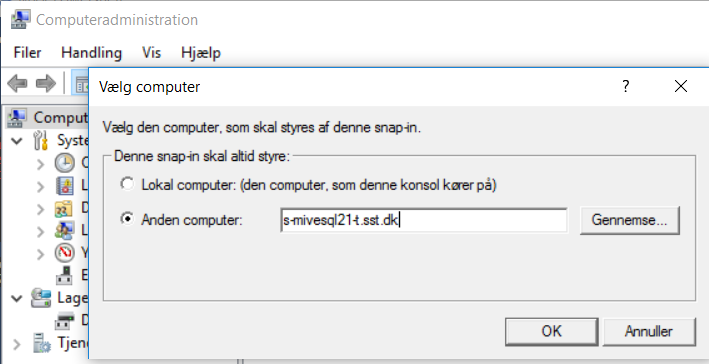
Brug ALTID SQL Configuration Manager (rigtig version)

Før SQL 2016 blev configuration Manager installeret sammen med SQL.

Efter SQL 2016 & Windows Core?: Log på arbejdsplads / server med Gui hvor der er installeret SQL server 2016 med adm\_bruger for pågældende domain

Start Computer Management (compmgmt.msc)

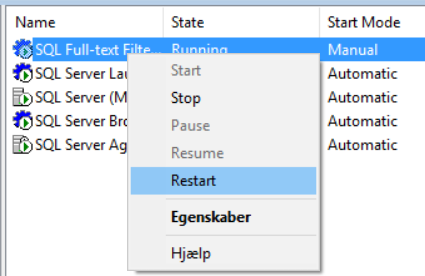
Højreklik Computer Management(Local Computer) vælg Connect to another computer.



Angiv destinations computer og tryk OK

Så fremkommer der en ny fane under Tjenester & programmer: SQL Server Configuration Manager

Vælg SQL Server Services, højreklik på service navn og vælg handling.



Ved stop: start med Integration, Analysis, Reporting -services, og slut med Engine

Hvis der er mere end 1 instans, luk de navngivne først og slut med MSSQLSERVER

Start services i omvendt rækkefølge.

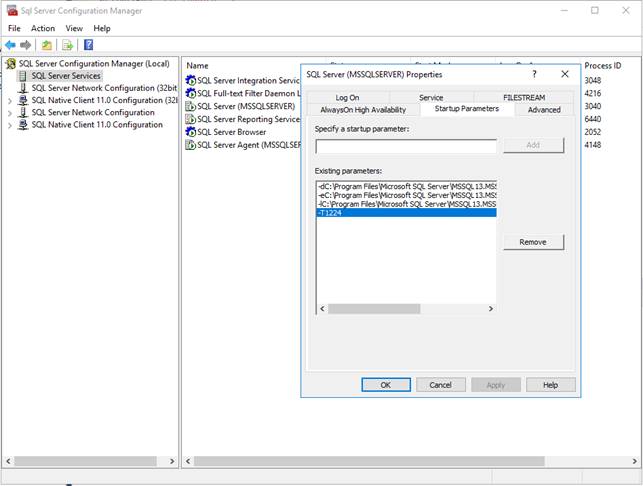
## Ret SQL opstarts parametre / Trace flags

Forbind til SQL Configuration Manager (rigtig version) se ovenstående

Så fremkommer der en ny fane under Tjenester & programmer: SQL Server Configuration Manager

Vælg SQL Server Services, højreklik på SQL Server (<engine instansnavn>) og vælg egenskaber.

Vælg StartupParameters, angiv Parameteren f.eks. –T1224 tryk Add



Denne indstilling deaktiverer låsning af eskalering baseret på antallet af låse og medvirker til at reducere CPU-forbruget på SQL Server 2016.

Tryk ok, og stop + Start engine for at aktivere.

## Oprettelse af MSSQL DBA login til SQL, hvis den mangler

USE [master];

CREATE LOGIN [DKSUNDTEST\l-org-adm-MsDbServer] FROM WINDOWS WITH DEFAULT\_DATABASE=[master];

EXEC master..sp\_addsrvrolemember @loginame = N'DKSUNDTEST\l-org-adm-MsDbServer', @rolename = N'sysadmin';

GO

## Opret Ny SQL Database

Use Master;

Create Database MinNyeDb

Sæt database Offline

ALTER DATABASE [myDB] SET OFFLINE WITH ROLLBACK IMMEDIATE;

## Slet database der er offline

USE [master];

declare @dbN sysname

Set @dbN ='mTIME\_STPS\_TEST' -- Angiv Databasenavn uden []

declare @db sysname

Set @db ='[@dbn]'

If Not Exists(select 1 from sys.databases where name = @dbN)

Select @dbn as DB, 'Findes ikke' as Status

Else

Begin

If Exists(select 1 from sys.databases where name = @dbN and state= 6)

Exec('ALTER DATABASE ' + @dbn + ' SET ONLINE');

If Exists(select 1 from sys.databases where name = @dbN and state= 0)

Begin

EXEC msdb.dbo.sp\_delete\_database\_backuphistory @database\_name = @dbN;

Exec('DROP DATABASE ' + @dbn)

End

If Exists(select 1 from sys.databases where name = @dbN)

select \* from sys.databases where name = @dbN

Else

Select @dbn as DB, 'Slettet' as Status

End

## DB\_Owner & anbefalinger

Det anbefales at alle fil størrelser sættes til fast værdi og ikke % relativt. F.eks datafil growth til 128MB og logfil growth til 64 MB.

Det anbefales at rette dbowner til SA

Det anbefales at sætte recovery til FULL.

[SQL Recovery Model: Simple vs. Full - DZone Database](https://dzone.com/articles/sql-recovery-model-simple-vs-full)

Declare @komd as nvarchar(max)

Set @komd='

If ''?'' not in (''master'',''tempdb'',''model'',''msdb'',''ssisdb'',''ReportServer'',''ReportServerTempDB'')

If((SELECT recovery\_model\_desc FROM master.sys.databases WHERE name = ''?'') !=''Full'')

Begin

USE [master] ;

ALTER DATABASE [?] SET RECOVERY FULL

Print ''?''

End

'

Exec sp\_MSforeachdb @komd;

SELECT suser\_sname(owner\_sid) db\_owner, \* FROM sys.databases -- viser db owner for alle db

Skift alle databaser på instans til owner=sa

-- EXEC sp\_MSforeachdb 'EXEC [?]..sp\_changedbowner ''sa''' --? kan misse databaser

SELECT 'ALTER AUTHORIZATION ON DATABASE::' + QUOTENAME(name) + ' TO [sa];'

from sys.databases

where name not in ('master', 'model', 'msdb ', 'tempdb')

/\* Via powershell \*/

#if you have named instance it would be \SQL\server\instance\Databases I believe cd MSSQLSERVER:\SQL\server\Databases

Set the owner for all objects in a directory

IR | foreach-object {$.SetOwner('sa'); $.Refresh()}

-- List jobnames & owner of all jobs

USE MSDB

GO

--SELECT GETDATE() AS 'ExecutionTime'

--GO

--SELECT @@SERVERNAME AS 'SQLServerInstance'

--GO

SELECT j.[name] AS 'JobName',

Enabled = CASE WHEN j.Enabled = 0 THEN 'No'

ELSE 'Yes'

END,

l.[name] AS 'OwnerName'

,l.dbname AS 'Database'

FROM MSDB.dbo.sysjobs j

-- INNER JOIN Master.dbo.syslogins l

Left JOIN Master.dbo.syslogins l

ON j.owner\_sid = l.sid

ORDER BY j.[name]

GO

### -- Change job ownership of single job to sa

EXEC MSDB.dbo.sp\_update\_job

@job\_name = 'DailyBackups',

@owner\_login\_name = 'sa'

GO

-- give others ownership of a job: In the query window, enter the following statements that use the

-- sp\_manage\_jobs\_by\_login (Transact-SQL) system stored procedure. The following example reassigns all jobs owned by danw giving new owner françoisa

USE msdb ;

GO

EXEC dbo.sp\_manage\_jobs\_by\_login

@action = N'REASSIGN',

@current\_owner\_login\_name = N'danw',

@new\_owner\_login\_name = N'françoisa' ;

GO

-- Conditionally build code to correct all Jobs not owned by the sa login then review the output before

-- executing the code in another SQL Server session to correct the SQL Server Agent Job ownership

SET NOCOUNT ON

SELECT 'EXEC MSDB.dbo.sp\_update\_job ' + char(13) +

'@job\_name = ' + char(39) + j.[Name] + char(39) + ',' + char(13) +

'@owner\_login\_name = ' + char(39) + 'sa' + char(39) + char(13) + char(13)

FROM MSDB.dbo.sysjobs j

-- INNER JOIN Master.dbo.syslogins l

LEFT JOIN Master.dbo.syslogins l

ON j.owner\_sid = l.sid

WHERE l.[name] <> 'sa'

or l.[name] is null -- orphan entries.

ORDER BY j.[name]

### --Vis alle Aktive kørende Jobs I SQL agent.

IF EXISTS (SELECT \* FROM tempdb.dbo.sysobjects

WHERE id = OBJECT\_ID(N'[tempdb].[dbo].[Temp1]') )

DROP TABLE [tempdb].[dbo].[Temp1]

GO

CREATE TABLE [tempdb].[dbo].[Temp1]

(job\_id uniqueidentifier NOT NULL

,last\_run\_date nvarchar (20) NOT NULL

,last\_run\_time nvarchar (20) NOT NULL

,next\_run\_date nvarchar (20) NOT NULL

,next\_run\_time nvarchar (20) NOT NULL

,next\_run\_schedule\_id INT NOT NULL

,requested\_to\_run INT NOT NULL

,request\_source INT NOT NULL

,request\_source\_id sysname COLLATE database\_default NULL

,running INT NOT NULL

,current\_step INT NOT NULL

,current\_retry\_attempt INT NOT NULL

,job\_state INT NOT NULL)

DECLARE @job\_owner sysname

DECLARE @is\_sysadmin INT

SET @is\_sysadmin = isnull (is\_srvrolemember ('sysadmin'), 0)

SET @job\_owner = suser\_sname ()

INSERT INTO [tempdb].[dbo].[Temp1]

--EXECUTE sys.xp\_sqlagent\_enum\_jobs @is\_sysadmin, @job\_owner

EXECUTE master.dbo.xp\_sqlagent\_enum\_jobs @is\_sysadmin, @job\_owner

UPDATE [tempdb].[dbo].[Temp1]

SET last\_run\_time = right ('000000' + last\_run\_time, 6),

next\_run\_time = right ('000000' + next\_run\_time, 6);

-----

SELECT j.name AS JobName

,j.enabled AS Enabled

,CASE x.running WHEN 1 THEN 'Running'

ELSE CASE h.run\_status

WHEN 2 THEN 'Inactive'

WHEN 4 THEN 'Inactive'

ELSE 'Completed'

END

END AS CurrentStatus

,coalesce (x.current\_step, 0) AS CurrentStepNbr

,CASE WHEN x.last\_run\_date > 0 THEN

convert (datetime, substring (x.last\_run\_date, 1, 4) + '-' + substring (x.last\_run\_date, 5, 2) + '-'

+ substring (x.last\_run\_date, 7, 2) + ' ' + substring (x.last\_run\_time, 1, 2) + ':'

+ substring (x.last\_run\_time, 3, 2) + ':' + substring (x.last\_run\_time, 5, 2) + '.000', 121)

ELSE NULL

END AS LastRunTime

,CASE h.run\_status

WHEN 0 THEN 'Fail'

WHEN 1 THEN 'Success'

WHEN 2 THEN 'Retry'

WHEN 3 THEN 'Cancel'

WHEN 4 THEN 'In progress'

END AS LastRunOutcome

,CASE WHEN h.run\_duration > 0 THEN

(h.run\_duration / 1000000) \* (3600 \* 24)

+ (h.run\_duration / 10000 % 100) \* 3600

+ (h.run\_duration / 100 % 100) \* 60

+ (h.run\_duration % 100)

ELSE NULL

END AS LastRunDuration

FROM [tempdb].[dbo].[Temp1] x

LEFT JOIN msdb.dbo.sysjobs j ON x.job\_id = j.job\_id

LEFT OUTER JOIN msdb.dbo.syscategories c ON j.category\_id = c.category\_id

LEFT OUTER JOIN msdb.dbo.sysjobhistory h ON x.job\_id = h.job\_id

AND x.last\_run\_date = h.run\_date

AND x.last\_run\_time = h.run\_time

AND h.step\_id = 0

where x.running = 1

-- eller tilsvarende:

SELECT ja.job\_id

,j.name AS job\_name

,ja.start\_execution\_date

,ISNULL(last\_executed\_step\_id,0)+1 AS current\_executed\_step\_id

,Js.step\_name

,js.last\_run\_outcome

FROM msdb.dbo.sysjobactivity ja

LEFT JOIN msdb.dbo.sysjobhistory jh ON ja.job\_history\_id = jh.instance\_id

JOIN msdb.dbo.sysjobs j ON ja.job\_id = j.job\_id

JOIN msdb.dbo.sysjobsteps js ON ja.job\_id = js.job\_id

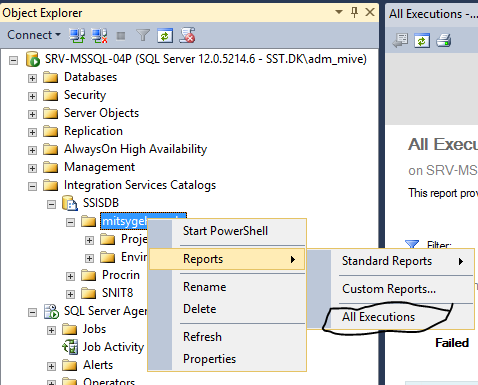
AND ISNULL(ja.last\_executed\_step\_id,0)+1 = js.step\_id

WHERE ja.session\_id = (SELECT TOP 1 session\_id FROM msdb.dbo.syssessions ORDER BY agent\_start\_date DESC)

AND start\_execution\_date is not null

AND stop\_execution\_date is null

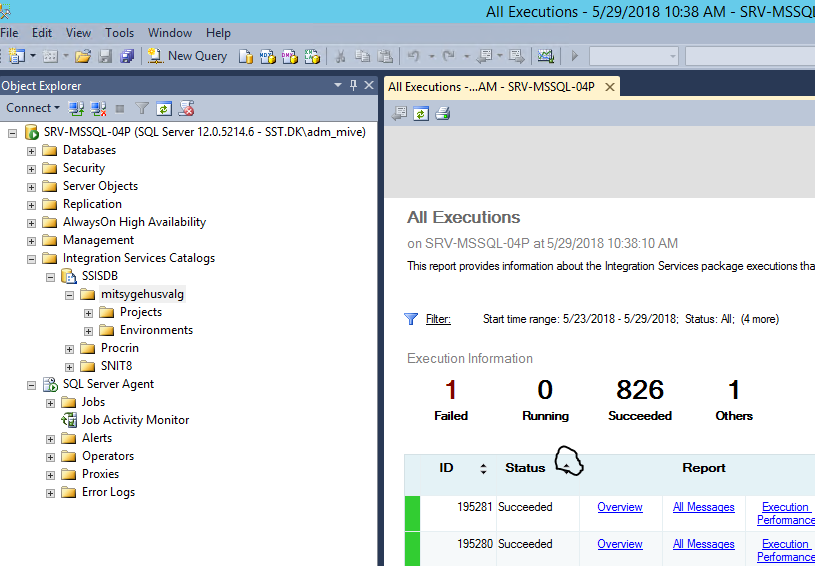
### How to stop a runaway SSIS package



På SQL Serveren

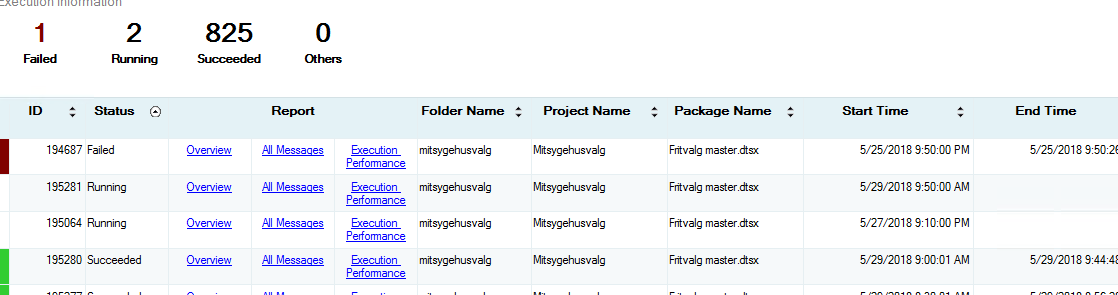
Vælg:

Integration Services Catalogs \ SSISDB \ <SSISProjekt> \ Reports \ All Executions

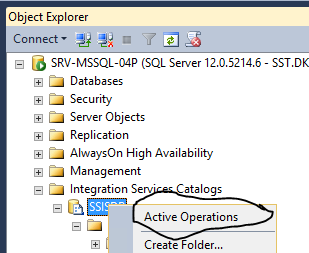


Her fremkommer liste over SSIS jobs

Hvis man klikker på pilOP ved status, kan man få sorteret så Failed & Running jobs vises først

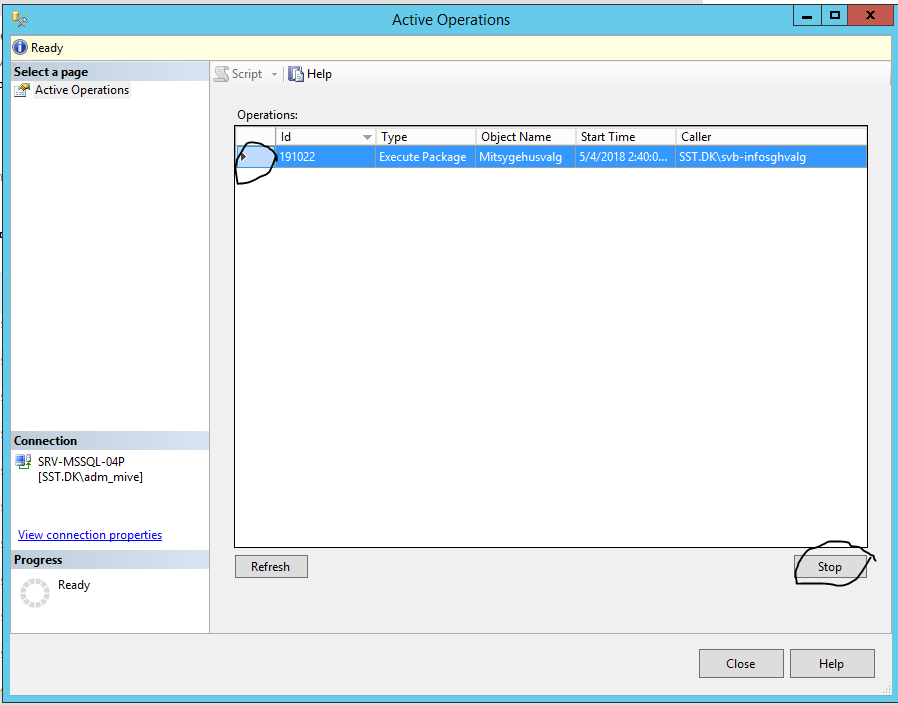


Her kan noteres ID for job der skal stoppes.

Så vælger man

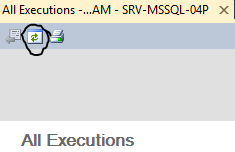
Integration Services Catalogs \ SSISDB \ Active Operations

Der kan godt gå meget lang tid.



Her skal vælges Job med samme ID som fejlede før, og tryk på Stop.

Det kan igen vare lang tid før der svares.



Tryk Refresh på all executions vinduet for at se det opdaterede resultat.

Og sorter evt. igen.

Yderligere info: https://www.sqlshack.com/stop-runaway-ssis-package/

# SQL Locks & Deadlock

Når man skal skrive i en database, låser man feltet (eller rækken, eller et område) for skrivning, og låser det op igen bagefter.

Når en anden proces forsøger at tilgå de låste data opstår der en Lock ( låst situation) og processen bliver sat i waiting on process.

Når den første proces bliver færdig, låser den op for de andre locks, der så automatisk fortsætter.

Deadlock er når 2 processer forsøger at tilgå og låse de samme ressourcer på samme tid og ikke kan løse sig selv. Det er umuligt helt at undgå deadlocks, man skal tage højde for dem ved fejlhåndtering.

MSSQL systemet vælger den process der er "billigst" og stopper denne, så de(n) andre processer kan fortsætte. "billigst" er en algoritme mssql udregner for at kunne vælge hvilken proces der skal slås ned.

SQL Programmøren skal tage højde for dette ved at indsætte checkpoints, med Commit / rollback, samt ved at låse(for skrivning) på så små datamængder som muligt. Det er bedre at låse på en række fremfor en tabel, og hvis man ikke angiver andet, kan man risikere at man låser på hele databasen.

## Procedure til søgning af deadlocks via T-Sql

--drop TABLE #temp\_sp\_who2

CREATE TABLE #temp\_sp\_who2

(

SPID INT,

Status VARCHAR(1000) NULL,

Login SYSNAME NULL,

HostName SYSNAME NULL,

BlkBy SYSNAME NULL,

DBName SYSNAME NULL,

Command VARCHAR(1000) NULL,

CPUTime INT NULL,

DiskIO BIGINT NULL, -- int

LastBatch VARCHAR(1000) NULL,

ProgramName VARCHAR(1000) NULL,

SPID2 INT

, RequestId INT NULL --comment out for SQL 2000 databases

)

INSERT INTO #temp\_sp\_who2

EXEC sp\_who2

SELECT \*

FROM #temp\_sp\_who2

WHERE DBName Like 'IB\_doedsaarsag%'

--kill xyz

## List antal Deadlock forekomster siden SQL engine start.

Se link: [How to report on SQL Server deadlock occurrences (sqlshack.com)](https://www.sqlshack.com/report-sql-server-deadlock-occurrences/)

Viser hvor mange deadlocks der har været siden start af SQLengine (virker)

SELECT

'Deadlocks Occurrences Report',

CONVERT(BIGINT,((1.0 \* p.cntr\_value /

NULLIF(datediff(DD,d.create\_date,CURRENT\_TIMESTAMP),0)))) as

AveragePerDay,

CAST(p.cntr\_value AS NVARCHAR(100)) + ' deadlocks have been recorded

since startup.' AS Details,

d.create\_date as StartupDateTime

FROM sys.dm\_os\_performance\_counters p

INNER JOIN sys.databases d ON d.name = 'tempdb'

WHERE RTRIM(p.counter\_name) = 'Number of Deadlocks/sec'

AND RTRIM(p.instance\_name) = '\_Total'

;

## Via Extended events

Serverinstance\Management\Extended events\Sessions\System\_Health

Husk Tools\Options \query results \SQL server \Results to Test \ Max number chars displayed: 9999

Andre links:

[Monitoring SQL Server Deadlocks - the easy way (sqlshack.com)](https://www.sqlshack.com/monitoring-sql-server-deadlocks-easy-way/)

[SQL SERVER - How to get historical deadlock Information from System Health Extended Events? - SQL Authority with Pinal Dave](https://blog.sqlauthority.com/2017/01/09/sql-server-get-historical-deadlock-information-system-health-extended-events/)

SELECT XEvent.query('(event/data/value/deadlock)[1]') AS DeadlockGraph

FROM (

SELECT XEvent.query('.') AS XEvent

FROM (

SELECT CAST(target\_data AS XML) AS TargetData

FROM sys.dm\_xe\_session\_targets st

INNER JOIN sys.dm\_xe\_sessions s ON s.address = st.event\_session\_address

WHERE s.NAME = 'system\_health'

--AND st.target\_name = 'ring\_buffer'

) AS Data

CROSS APPLY TargetData.nodes('RingBufferTarget/event[@name="xml\_deadlock\_report"]') AS XEventData(XEvent)

) AS source;

DBCC Trace(1222)

enable trace flag 1222

## Begræns Memory Used

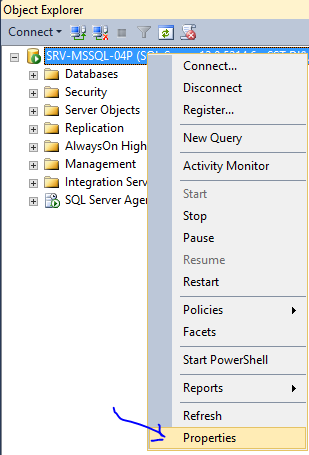
På server find memory used:

Find pagefile størrelse

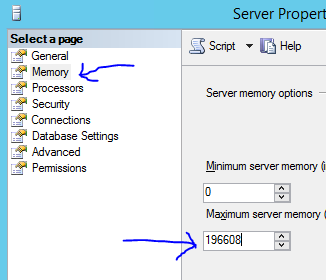
Vælg en fornuftig størrelse der skal reserveres til (hver) SQL instans

Åbn SQL Instans i MS SQL Server Management Studio

Højreklik SQL Instans og vælg Properties.



Vælg Memory, og Angiv Maximum Server Memory for denne instans.



Tryk Ok for at afslutte tildelingen

Det burde kunne ses indenfor 10 min i Check\_MK

### Find SQL Recovery Model, Last Backup & Backup type

Dette script giver et hurtigt overblik over databasers recovery\_model og deres seneste LOG backup.

USE MSDB

GO

Select @@SERVERNAME as SqlInstance, B.database\_name, type, max(B.backup\_finish\_date) as backup\_finish\_date

,MAX(D.recovery\_model\_desc) as recovery\_model\_desc

,MAX(D.log\_reuse\_wait\_desc) as log\_reuse\_wait\_desc

from backupset B

JOIN sys.databases D ON D.name = B.database\_name

Where B.type='L' --her kan du bare ændre til fx D (D = database/full backup)

or B.type='D'

Group by B.database\_name, type

Order by B.database\_name, type

# Oversigt over Extended events.

Extended events bruges til at logge hvad der sker i en SQL database / server, med det formål at analysere data bagefter.

Der er pt. 15 pakker med extended events, der kan findes via:

select \*

from sys.dm\_xe\_packages p

name description

package0 Default package. Contains all standard types, maps, compare operators, actions and targets

sqlos Extended events for SQL Operating System

sqlserver Extended events for Microsoft SQL Server

SecAudit Security Audit Events

ucs Extended events for Unified Communications Stack

sqlclr Extended events for SQL CLR

filestream Extended events for SQL Server FILESTREAM and FileTable

sqlserver Extended events for Microsoft SQL Server

SQLSatellite Extended events for SQL Satellite

sqlsni Extended events for Microsoft SQL Server

sqlserver Extended events for Microsoft SQL Server

qds Extended events for Query Store

XtpRuntime Extended events for the XTP Runtime

XtpCompile Extended events for the XTP Compile

XtpEngine Extended events for the XTP Engine

Af disse kan vurderes at

Extended events for SQL Operating System

Extended events for Microsoft SQL Server

Extended events for Unified Communications Stack

Extended events for SQL CLR

Extended events for SQL Server FILESTREAM and FileTable

Extended events for Microsoft SQL Server

Extended events for SQL Satellite

Extended events for Microsoft SQL Server

Extended events for Microsoft SQL Server

Extended events for Query Store

Extended events for the XTP Runtime

Extended events for the XTP Compile

Extended events for the XTP Engine

IKKE har nogen relation til med Data, (mere specifikt GDPR data), men mere opsætnings parametre, system & ydre enheder

Default package. Contains all standard types, maps, compare operators, actions and targets

Security Audit Events

select p.name package\_name

, p.description package\_description

, o.name event\_name

, o.description event\_description

from sys.dm\_xe\_objects o

join sys.dm\_xe\_packages p

on o.package\_guid = p.guid

--where p.name in ('package0','SecAudit') --and o.object\_type = 'event'

--where o.name in ('sql\_batch\_completed','rpc\_completed')

order by package\_name

# Audit Logning fra Extended events.

CREATE EVENT SESSION [AuditLog] ON SERVER

ADD EVENT sqlserver.rpc\_completed(SET collect\_statement=(1)

ACTION(sqlserver.client\_app\_name,sqlserver.client\_hostname,sqlserver.database\_name,sqlserver.server\_instance\_name,sqlserver.session\_id,sqlserver.username)

WHERE ([package0].[greater\_than\_uint64]([sqlserver].[database\_id],(4)) AND [package0].[equal\_boolean]([sqlserver].[is\_system],(0)) AND NOT ([sqlserver].[like\_i\_sql\_unicode\_string]([sqlserver].[username],N'%svt\_%') OR [sqlserver].[like\_i\_sql\_unicode\_string]([sqlserver].[username],N'%svb\_%') OR [sqlserver].[like\_i\_sql\_unicode\_string]([sqlserver].[username],N'%svd\_%') OR [sqlserver].[like\_i\_sql\_unicode\_string]([sqlserver].[username],N'%$') OR [sqlserver].[like\_i\_sql\_unicode\_string]([sqlserver].[username],N'NT SERVICE%') OR [sqlserver].[like\_i\_sql\_unicode\_string]([sqlserver].[username],N'NT AUTHORITY%') OR [sqlserver].[like\_i\_sql\_unicode\_string]([sqlserver].[username],N'%svk-%') OR [sqlserver].[like\_i\_sql\_unicode\_string]([sqlserver].[username],N'%svk\_%')))),

ADD EVENT sqlserver.sql\_batch\_completed(SET collect\_batch\_text=(1)

ACTION(sqlserver.client\_app\_name,sqlserver.client\_hostname,sqlserver.database\_name,sqlserver.server\_instance\_name,sqlserver.session\_id,sqlserver.username)

WHERE ([package0].[greater\_than\_uint64]([sqlserver].[database\_id],(4)) AND [package0].[equal\_boolean]([sqlserver].[is\_system],(0)) AND NOT ([sqlserver].[like\_i\_sql\_unicode\_string]([sqlserver].[username],N'%svt\_%') OR [sqlserver].[like\_i\_sql\_unicode\_string]([sqlserver].[username],N'%svb\_%') OR [sqlserver].[like\_i\_sql\_unicode\_string]([sqlserver].[username],N'%svd\_%') OR [sqlserver].[like\_i\_sql\_unicode\_string]([sqlserver].[username],N'%$') OR [sqlserver].[like\_i\_sql\_unicode\_string]([sqlserver].[username],N'NT SERVICE%') OR [sqlserver].[like\_i\_sql\_unicode\_string]([sqlserver].[username],N'NT AUTHORITY%') OR [sqlserver].[like\_i\_sql\_unicode\_string]([sqlserver].[username],N'%svk-%') OR [sqlserver].[like\_i\_sql\_unicode\_string]([sqlserver].[username],N'%svk\_%'))))

ADD TARGET package0.event\_file(SET filename=N'S:\System\MSSQL13.MSSQLSERVER\MSSQL\AuditEE\AuditEE.xel',max\_file\_size=(100),max\_rollover\_files=(100))

WITH (MAX\_MEMORY=4096 KB,EVENT\_RETENTION\_MODE=ALLOW\_SINGLE\_EVENT\_LOSS,MAX\_DISPATCH\_LATENCY=30 SECONDS,MAX\_EVENT\_SIZE=0 KB,MEMORY\_PARTITION\_MODE=NONE,TRACK\_CAUSALITY=OFF,STARTUP\_STATE=ON)

GO

ISJSON ( ExtendedEvent )

SELECT TOP (10) JSON\_QUERY (ExtendedEvent) as String

FROM [AuditLog].[dbo].[AuditLogV2]

The key difference between **JSON\_VALUE** and **JSON\_QUERY** is that **JSON\_VALUE** returns a scalar value, while **JSON\_QUERY** returns an object or an array.

/\*\*\*\*\*\* Script for SelectTopNRows command from SSMS \*\*\*\*\*\*/

SELECT TOP (100) json\_value(ExtendedEvent, '$.Name') as Navn

,json\_value(ExtendedEvent, '$.UUID') as UUID

,json\_value(ExtendedEvent, '$.Timestamp') as Timestamp

,json\_value(ExtendedEvent, '$.Fields.cpu\_time') as cpu\_time

,json\_value(ExtendedEvent, '$.Fields.duration') as duration

,json\_value(ExtendedEvent, '$.Fields.physical\_reads') as physical\_reads

,json\_value(ExtendedEvent, '$.Fields.logical\_reads') as logical\_reads

,json\_value(ExtendedEvent, '$.Fields.writes') as writes

,json\_value(ExtendedEvent, '$.Fields.spills') as spills

,json\_value(ExtendedEvent, '$.Fields.row\_count') as row\_count

,json\_value(ExtendedEvent, '$.Fields.result') as result

,json\_value(ExtendedEvent, '$.Fields.batch\_text') as batch\_text

,json\_value(ExtendedEvent, '$.Fields.connection\_reset\_option') as connection\_reset\_option

,json\_value(ExtendedEvent, '$.Fields.object\_name') as object\_name

,json\_value(ExtendedEvent, '$.Fields.statement') as statement

,json\_value(ExtendedEvent, '$.Fields.data\_stream') as data\_stream

,json\_value(ExtendedEvent, '$.Fields.output\_parameters') as output\_parameters

,json\_value(ExtendedEvent, '$.Actions.session\_id') as session\_id

,json\_value(ExtendedEvent, '$.Actions.server\_instance\_name') as server\_instance\_name

,json\_value(ExtendedEvent, '$.Actions.username') as username

,json\_value(ExtendedEvent, '$.Actions.database\_name') as database\_name

,json\_value(ExtendedEvent, '$.Actions.client\_hostname') as client\_hostname

,json\_value(ExtendedEvent, '$.Actions.client\_app\_name') as client\_app\_name

,ExtendedEvent

FROM [AuditLog].[dbo].[AuditLogv2]

WHERE ISJSON(ExtendedEvent) > 0

[Work with JSON data - SQL Server | Microsoft Docs](https://docs.microsoft.com/en-us/sql/relational-databases/json/json-data-sql-server?view=sql-server-ver15)

## Gem Check\_MK databaseliste I tabel

Insert into check\_MK\_Fetch.dbo.db

SELECT Cast(@@SERVERNAME+'\'+DEFAULT\_DOMAIN()+'\'+@@servicename as nvarchar) as FQInstanceName

,SQLVersion=Left(@@VERSION,Charindex('(',@@VERSION)-1)

,Cast(SERVERPROPERTY('productversion') as nvarchar) as 'Version'

,Cast(SERVERPROPERTY('productlevel') as nvarchar) as ServicePack

,Cast(SERVERPROPERTY('edition') as nvarchar) as SQLEdition

,DB\_NAME(database\_id) AS DbName

,Datasize=(select sum(Cast(x.size as Bigint)\*8)/1024 FROM sys.master\_files x where x.database\_id=m.database\_id and x.type=0)

,DataGrowth=(select sum(x.growth\*8)/1024 FROM sys.master\_files x where x.database\_id=m.database\_id and x.type=0)

,Logsize=(select sum(Cast(x.size as Bigint)\*8)/1024 FROM sys.master\_files x where x.database\_id=m.database\_id and x.type=1)

,LogGrowth=(select sum(x.growth\*8)/1024 FROM sys.master\_files x where x.database\_id=m.database\_id and x.type=1)

,LastFullBk=(select max(backup\_finish\_date) FROM msdb.dbo.backupset x where DB\_NAME(m.database\_id) = x.database\_name and x.type='D')

,LastLogBk=(select max(backup\_finish\_date) FROM msdb.dbo.backupset x where DB\_NAME(m.database\_id) = x.database\_name and x.type='L')

,GetDate() as logtid

FROM sys.master\_files m

where DB\_NAME(database\_id) not in ('master','tempdb','model','msdb','ReportServer','ReportServerTempDB')

group by database\_id

## Hent nyeste Check\_MK databaseliste fra tabel

SELECT dbname,datasize,datagrowth,logsize,loggrowth,LastFullbk,LastLogbk,logtid

FROM Check\_MK\_fetch.dbo.DB

where logtid=(select max(logtid) FROM Check\_MK\_fetch.dbo.DB)

## Denne del danner tekst strengen til Check\_mk (nedenstående er med et dårlige eksempel)

Declare @Dumpline AS Nvarchar(MAX)

SELECT @Dumpline = COALESCE(@Dumpline + ' ' , '') + db.name + ' ' + COALESCE(@Dumpline + ' ', '') + db.filename + ' ' + COALESCE(@Dumpline + CHAR(13)+CHAR(10), '')

FROM master.dbo.sysdatabases db

print @Dumpline ;

## Hent SQL tekst strengen til Check\_mk

SELECT dbBlob, logtid

FROM Check\_MK\_fetch.dbo.Blob

## Check om Kolonne findes I Tabel.

If (Exists(select \* from INFORMATION\_SCHEMA.COLUMNS

WHERE TABLE\_CATALOG ='DBA\_DB'

And TABLE\_SCHEMA = 'dbo'

AND TABLE\_NAME = 'InstanceInfo'

AND COLUMN\_NAME='LogicalCPU'))

Begin … End

## SQL Adgang

### Standard opsætning med Læs, Skriv, dbo grupper.

Der oprettes som standard 3 stk. Globale Windows AD brugerprivilegiegrupper til 1 (eller flere sammenhængende) database(r).

Servicedesk melder brugere ind i disse G-grupper.

De skal følge navnestandarden: <system>\_g\_<domæne>\_<databasenavn>\_<r|w|o>

Disse G-brugergrupper meldes ind i de tilhørende Lokale ressource grupper:

<system>\_l\_<domæne>\_<databasenavn>\_<r|w|o>

Disse L-ressourcegrupper giver adgang til & privilegier på ressourcer som diske & databaser.

Specifikt oprettes der MSSQL Instans Logins, der tilknyttes L-ressourcegrupper, således at der gives adgang til SQL Server instansen via AD-grupperne.

Der oprettes tillige Usere på database niveau, der knyttes sammen med førnævnte Instans logins

## SQL Login – adgang til SQL server Instans

USE [master]

GO

CREATE LOGIN [DKSUND\MIVE] FROM WINDOWS WITH DEFAULT\_DATABASE=[master]

GO

## DB User Tilknyt Login til DataBase (kræver Login)

USE [mive\_testDB]

GO

CREATE USER [Dksund\mive] FOR LOGIN [DKSUND\mive] WITH DEFAULT\_SCHEMA=[dbo]

GO

## DB Privilegier til data på DataBase (kræver Login & bruger oprettet på db)

USE [mive\_testDB]

GO

ALTER ROLE [db\_datawriter] ADD MEMBER [dksund\mive]

GO

## DB Privilegier på view (kræver Login & bruger oprettet på db)

USE [mive\_testDB]

GO

GRANT SELECT ON [dbo].[LTestView] TO [Dksund\mive]

GO

## Opret SQL Sysadmin login Emergency

USE [master]

GO

CREATE LOGIN [MiveBack] WITH PASSWORD=N'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*', DEFAULT\_DATABASE=[master], CHECK\_EXPIRATION=OFF, CHECK\_POLICY=ON

GO

ALTER SERVER ROLE [sysadmin] ADD MEMBER [MiveBack]

GO

Erstat \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* af kendt kompliceret pw, gem I SecretServer

Kan bruges hvis man skal flytte / kopiere server til nyt domain

<https://docs.microsoft.com/en-us/sql/relational-databases/security/authentication-access/database-level-roles>

P:\anjeT2\DBA\Tools\sp\_help\_revlogin.txt

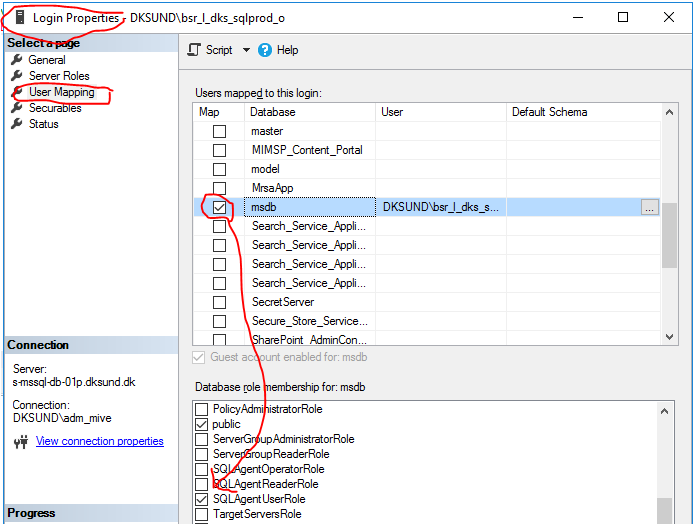
SQL Server Backup Strategi

### Styring af SQL agent jobs & roller

Hvis man skal kunne se resultat af SQL Jobs, skal man have adgang til SQL Agent Job køen.

Det kræver at man får tildelt en SQL Agent rolle.

På databasen MSDN tilknyttes Instans Login med rollen: SQLAgentUserRole



<https://docs.microsoft.com/en-us/sql/ssms/agent/sql-server-agent-fixed-database-roles>

de 2 andre agent roller giver mere adgang til at rette og schedulere andre systemers jobs, og bør kun tildeles på eksklusive SQL servere ALDRIG på SQL Hoteller.

## Se hvem der har SQLagentUserRole roller mm.

use msdb

Go

SELECT

roles.[name] as role\_name,

members.[name] as user\_name

FROM sys.database\_role\_members

JOIN sys.database\_principals roles ON database\_role\_members.role\_principal\_id = roles.principal\_id

JOIN sys.database\_principals members ON database\_role\_members.member\_principal\_id = members.principal\_id

ORDER BY

roles.[name],

members.[name]

## Activity Monitor tool - Statistics

requires VIEW SERVER STATE permission; this permission is required to select from DMVs such as sys.sysprocesses.  Open a connection (i.e. using sqlcmd or Management Studio) using your sysadmin credentials and issue the following statement:

**GRANT VIEW SERVER STATE TO << login name >>**

Or you can do it through SSMS.

1. Right Click on the SQL Login in SSMS under Security -> Logins and click Properties.
2. Click on Securables.
3. (Click “server 'Servername\Instance' ")
4. Eller
5. (Click Add
6. Select the last Radio Button that should be "The server 'Servername\Instance' "
7. Click Ok.)
8. The bottom box will be populated with Explicit permissions for ServerName\Instance
9. Scroll to the bottom of this list.
10. Check the box for *Grant* for **View Server State** Permission.

## Database dm Sys.dm\_db Database Statistikker

Use >database>

Go

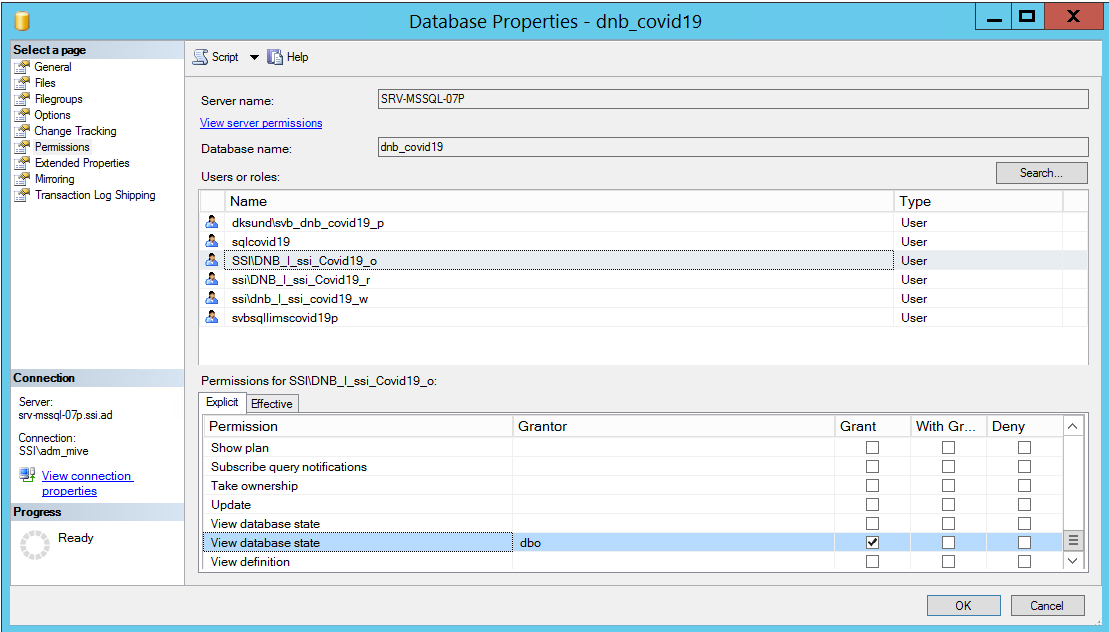
grant VIEW DATABASE STATE to [<ad\group>]

GO

to access statistics about the database.

Eller via SSMS

1. Højreklik database, vælg properties.
2. Permissions
3. Vælg burger (midt I = Users or roles)
4. Permissions (for neden) find “View database state” sæt hak I Grant, tryk OK



## Se hvem der har VIEW SERVER STATE privilegier

use master;

set nocount on

select 'grantee' = ssprin.[name]

, 'state\_desc' = ssperm.[state\_desc]

, 'permission' = ssperm.[permission\_name]

from sys.server\_permissions ssperm

join sys.server\_principals ssprin

on ssperm.grantee\_principal\_id = ssprin.principal\_id

where ssperm.[permission\_name] in ('view server state', 'view any database', 'view any definition')

and ssprin.[name] not in (

'sa', 'NT SERVICEWinmgmt', 'NT SERVICEMSSQLSERVER', 'NT AUTHORITYSYSTEM', 'NT SERVICESQLWriter'

, '##MS\_SQLAuthenticatorCertificate##', '##MS\_AgentSigningCertificate##', '##MS\_PolicyEventProcessingLogin##'

, '##MS\_PolicySigningCertificate##', '##MS\_PolicyTsqlExecutionLogin##', '##MS\_SmoExtendedSigningCertificate##'

, '##MS\_SQLResourceSigningCertificate##', '##MS\_SQLReplicationSigningCertificate##'

)

order by ssperm.[permission\_name] asc

### Login Errors (18456)

A login failure will throw an 18456 error and will be accompanied by an entry in the SQL Server error log

The severity of the error indicates the seriousness of the error. A severity level of 14 indicates an error in the range described as user correctable, which is understandable for login failures. The next item of information the error provides is the state number. Most errors have a state number associated with them which provides further information which is usually unique to the error that has been thrown. For a login error, state: 8, shown in the above example, indicates an invalid password was used. The state number therefore provides invaluable information about the reason for the login failure and can often be enough to identify the cause of an 18456 error.

The table below illustrates what some of these state values mean:

|  |  |
| --- | --- |
| **State** | **Description** |
| 1 | Account is locked out |
| 2 | User id is not valid |
| 3-4 | Undocumented |
| 5 | Login not found; sometimes a connecting application is incorrectly set to specify connection via SQL Authentication for a Windows account |
| 6 | Undocumented |
| 7 | The login being used is disabled |
| 8 | Incorrect password |
| 9 | Invalid password |
| 10 | Related to a SQL login being bound to Windows domain password policy enforcement. See KB925744. |
| 11-12 | Login valid but server access failed |
| 16 | Login valid, but not permissioned to use the target database |
| 18 | Password expired |
| 27 | Initial database could not be found |
| 38 | Login valid but database unavailable (or login not permissioned) |

### Find Server rolle medlemmer

SELECT sys.server\_role\_members.role\_principal\_id, role.name AS RoleName,

sys.server\_role\_members.member\_principal\_id, member.name AS MemberName

FROM sys.server\_role\_members

JOIN sys.server\_principals AS role

ON sys.server\_role\_members.role\_principal\_id = role.principal\_id

JOIN sys.server\_principals AS member

ON sys.server\_role\_members.member\_principal\_id = member.principal\_id;

ALTER SERVER ROLE [sysadmin] ADD MEMBER [dksund\Calibry\_l\_dks\_Calibry\_o]

ALTER SERVER ROLE [sysadmin] DROP MEMBER [dksund\Calibry\_l\_dks\_Calibry\_o]

### find Instance Logins with default DB other than master & msdb

select name as LoginName

,principal\_id

,is\_disabled

,default\_database\_name as DefaultDB

from sys.server\_principals

where default\_database\_name not in ('master','msdb')

order by default\_database\_name

### find Instance Logins with more than user access. Shows Sysadmins

/\* find Instance Logins with more than user access. Shows Sysadmins \*/

select l.sid

,l.status

,l.accdate as "Last Access"

,l.name

,l.loginname

,l.dbname as "default DB"

,l.denylogin

,l.hasaccess

,l.sysadmin

,l.securityadmin

,l.serveradmin

,l.setupadmin

,l.processadmin

,l.diskadmin

,l.dbcreator

,l.bulkadmin

,l.isntname

,l.isntgroup

,l.isntuser

from master.dbo.syslogins as l

where l.loginname not in('NT SERVICE\MSSQLSERVER','NT SERVICE\SQLSERVERAGENT','NT AUTHORITY\SYSTEM','NT AUTHORITY\LOCAL SERVICE')

and

(l.sysadmin > '0' or l.securityadmin > '0' or l.serveradmin > '0' or l.setupadmin > '0' or l.processadmin > '0' or l.diskadmin > '0' or l.dbcreator > '0' or l.bulkadmin > '0')

--(l.sysadmin < '1' and l.securityadmin < '1' and l.serveradmin < '1' and l.setupadmin < '1' and l.processadmin < '1' and l.diskadmin < '1' and l.dbcreator < '1' and l.bulkadmin < '1')

### find DB users with more than user access. Shows db\_owner

/\* find DB users with more than user access. Shows db\_owner \*/

Select q.member\_name as DBUserName

,u.Hasdbaccess as DBadgang

,SUM(q.db\_owner) as db\_owner

,SUM(q.db\_securityadmin)as db\_securityadmin

,SUM(q.db\_accessadmin)as db\_accessadmin

,SUM(q.db\_ddladmin) as db\_ddladmin

,SUM(q.db\_backupoperator) as db\_backupoperator

,SUM(q.db\_datawriter) as db\_datawriter

,SUM(q.db\_datareader) as db\_datareader

,SUM(q.db\_denydatawriter)as db\_denydatawriter

,SUM(q.db\_denydatareader)as db\_denydatareader

from (

select m.name as member\_name

,r.name as role\_name

,case when r.name= 'db\_owner' then 1 else 0 end as db\_owner

,case when r.name= 'db\_securityadmin' then 1 else 0 end as db\_securityadmin

,case when r.name= 'db\_accessadmin' then 1 else 0 end as db\_accessadmin

,case when r.name= 'db\_ddladmin' then 1 else 0 end as db\_ddladmin

,case when r.name= 'db\_backupoperator' then 1 else 0 end as db\_backupoperator

,case when r.name= 'db\_datawriter' then 1 else 0 end as db\_datawriter

,case when r.name= 'db\_datareader' then 1 else 0 end as db\_datareader

,case when r.name= 'db\_denydatawriter' then 1 else 0 end as db\_denydatawriter

,case when r.name= 'db\_denydatareader' then 1 else 0 end as db\_denydatareader

from sys.database\_role\_members rm

inner join sys.database\_principals r on rm.role\_principal\_id = r.principal\_id

inner join sys.database\_principals m on rm.member\_principal\_id = m.principal\_id

) as q

Join dbo.sysusers as u

on q.member\_name=u.name

where q.db\_owner>0 or q.db\_securityadmin>0 or q.db\_accessadmin>0 or q.db\_ddladmin>0 or q.db\_backupoperator>0

group by q.member\_name,u.Hasdbaccess

--order by q.member\_name

### Find DB users with only user access

/\* find DB users with only user access \*/

Select q.member\_name as DBUserName

,u.Hasdbaccess as DBadgang

,SUM(q.db\_owner) as db\_owner

,SUM(q.db\_securityadmin)as db\_securityadmin

,SUM(q.db\_accessadmin)as db\_accessadmin

,SUM(q.db\_ddladmin) as db\_ddladmin

,SUM(q.db\_backupoperator) as db\_backupoperator

,SUM(q.db\_datawriter) as db\_datawriter

,SUM(q.db\_datareader) as db\_datareader

,SUM(q.db\_denydatawriter)as db\_denydatawriter

,SUM(q.db\_denydatareader)as db\_denydatareader

from (

select m.name as member\_name

,r.name as role\_name

,case when r.name= 'db\_owner' then 1 else 0 end as db\_owner

,case when r.name= 'db\_securityadmin' then 1 else 0 end as db\_securityadmin

,case when r.name= 'db\_accessadmin' then 1 else 0 end as db\_accessadmin

,case when r.name= 'db\_ddladmin' then 1 else 0 end as db\_ddladmin

,case when r.name= 'db\_backupoperator' then 1 else 0 end as db\_backupoperator

,case when r.name= 'db\_datawriter' then 1 else 0 end as db\_datawriter

,case when r.name= 'db\_datareader' then 1 else 0 end as db\_datareader

,case when r.name= 'db\_denydatawriter' then 1 else 0 end as db\_denydatawriter

,case when r.name= 'db\_denydatareader' then 1 else 0 end as db\_denydatareader

from sys.database\_role\_members rm

inner join sys.database\_principals r on rm.role\_principal\_id = r.principal\_id

inner join sys.database\_principals m on rm.member\_principal\_id = m.principal\_id

) as q

Join dbo.sysusers as u

on q.member\_name=u.name

where q.db\_owner <1 and q.db\_securityadmin <1 and q.db\_accessadmin<1 and q.db\_ddladmin<1 and q.db\_backupoperator<1

group by q.member\_name, u.Hasdbaccess

--order by q.member\_name

### Find db users who are connected to Instance Login via sid

/\* find db users who are connected to Instance Login via sid \*/

Select u.name

--,u.uid

--,u.status

,u.sid

--,u.altuid

--,u.gid

,u.islogin

,u.hasdbaccess

,u.isntname

,u.isntgroup

,u.isntuser

,u.issqluser

,u.isaliased

--,u.issqlrole

,u.isapprole

from dbo.sysusers as u

where u.sid in(select l.sid from master.dbo.syslogins as l)

and u.issqlrole<>1

order by u.isntgroup desc,u.isntuser desc,u.uid

### Find orphaned db users with broken links

/\* find db users with broken links, who are NOT connected to Instance Login via sid \*/

select \*

from sys.database\_principals -- db users

where type in ('S','U','G')

And sid not in ( Select Sid from sys.server\_principals) -- Instance logins

/\* find db users with broken links, who are NOT connected to Instance Login via sid \*/

Select u.name

--,u.uid

--,u.status

,u.sid

--,u.altuid

--,u.gid

,u.islogin

,u.hasdbaccess

,u.isntname

,u.isntgroup

,u.isntuser

,u.issqluser

,u.isaliased

--,u.issqlrole

,u.isapprole

from dbo.sysusers as u

where u.sid not in(select l.sid from master.dbo.syslogins as l)

And u.name not in ('guest')

and u.issqlrole<>1

order by u.uid

### Slet orphaned db user after migration

use Forsker;

declare @sql1 nvarchar(max)

set @sql1 = ''

SELECT @sql1 = @sql1+

'

print ''Deleting orphaned db user: '+name+'''

DROP USER ['+name+'];

'

FROM

dbo.sysusers as u

where u.sid not in(select l.sid from master.dbo.syslogins as l)

And u.name not in ('guest')

and u.issqlrole<>1

order by u.uid

Print @sql1

execute (@sql1)

### Tilknyt orphaned db user til instance login.

USE [master];

CREATE LOGIN [<domain>\<ad-group>] FROM WINDOWS WITH DEFAULT\_DATABASE=[<db>]; -- opret login

USE [<db>];

ALTER USER [<domain>\<ad-group>] WITH LOGIN = [<domain>\<ad-group>]; -- tilknyt db-user til login.

/\* opret nyt SQLlogin med oprindeligt Security-ID \*/

USE [master];

CREATE LOGIN [<domain>\<ad-group>] With SID= <sid-fra-dbuser>, Password= <password-skal-matche>

### Change all databases to Recovery Full

DECLARE @Database VARCHAR(255)

DECLARE @cmd NVARCHAR(500)

DECLARE DatabaseCursor CURSOR FOR

SELECT name FROM master.dbo.sysdatabases

WHERE name NOT IN ('master','msdb','tempdb','model')

ORDER BY 1

OPEN DatabaseCursor

FETCH NEXT FROM DatabaseCursor INTO @Database

WHILE @@FETCH\_STATUS = 0

BEGIN

SET @cmd = 'ALTER DATABASE ' + @Database + ' SET RECOVERY FULL WITH NO\_WAIT;';

print @cmd ;

EXEC (@cmd) ;

FETCH NEXT FROM DatabaseCursor INTO @Database

END

CLOSE DatabaseCursor

DEALLOCATE DatabaseCursor

### Change all tables in all databases to ?

DECLARE @Database VARCHAR(255)

DECLARE @Table VARCHAR(255)

DECLARE @cmd NVARCHAR(500)

DECLARE DatabaseCursor CURSOR FOR

SELECT name FROM master.dbo.sysdatabases

WHERE name NOT IN ('master','msdb','tempdb','model')

ORDER BY 1

OPEN DatabaseCursor

FETCH NEXT FROM DatabaseCursor INTO @Database

WHILE @@FETCH\_STATUS = 0

BEGIN

SET @cmd = 'DECLARE TableCursor CURSOR FOR SELECT ''['' + table\_catalog + ''].['' + table\_schema + ''].['' +

table\_name + '']'' as tableName FROM [' + @Database + '].INFORMATION\_SCHEMA.TABLES

WHERE table\_type = ''BASE TABLE'''

-- create table cursor

EXEC (@cmd)

OPEN TableCursor

FETCH NEXT FROM TableCursor INTO @Table

WHILE @@FETCH\_STATUS = 0

BEGIN

-- SQL 2005 or higher command

--SET @cmd = 'dbcc checktable (' + @Table + ') WITH DATA\_PURITY;';

--SET @cmd = ' UPDATE STATISTICS ' + @Table + ' WITH FULLSCAN;';

SET @cmd = 'ALTER INDEX ALL ON ' + @Table + ' REBUILD Partition = All;'; -- Uden replace

--SET @cmd =REPLACE(@cmd,'([','(''')

--SET @cmd =REPLACE(@cmd,'])',''')')

--SET @cmd =REPLACE(@cmd,'].[','.')

print @cmd ;

EXEC (@cmd) ;

FETCH NEXT FROM TableCursor INTO @Table

END

CLOSE TableCursor

DEALLOCATE TableCursor

FETCH NEXT FROM DatabaseCursor INTO @Database

END

CLOSE DatabaseCursor

DEALLOCATE DatabaseCursor

### Allow & Deny SQL Users & Logins

Use master

Go

Alter Login eks\_user disable;

Alter Login “ssi.ad\eks\_user”;

Use master

Go

Alter Login eks\_user enable;

Use aut\_xfer\_ssi

Go

Grant Connect to "eks\_user"

use aut\_xfer\_ssi

Go

Revoke Connect from “eks\_user”;

Revoke Connect from “ssi.ad\eks\_user”;

### Tillad execute på stored procedure:

Create Role db\_executor

Grant Execute to db\_executor

Eller

USE [SorXMLStaging];

Grant Execute to [SSIDMZ01\bsr\_l\_sql\_w]; -- skal være i [] hvis ikke SQL user.

-- giver ad-gruppe privilegier til at køre stored procedurer

Eller

USE [SorXMLStaging];

ALTER ROLE [db\_ddladmin] ADD MEMBER [SSIDMZ01\bsr\_l\_sql\_w]; -- skal være i [] hvis ikke SQL user.

-- mere beregnet til en Udvikler profil, kan køre alle DataDefinitionLanguage kommando i en db.

Grant Execute on Schema ::dbo to [SSIDMZ01\bsr\_l\_sql\_w]; -- hvis der skal begrænses til et schema.

use master

go

sp\_addlogin [domain\username]

sp\_addsrvrolemember [domain\username],**'sysadmin'**

-- sp\_addlogin is for SQL Authentication. You must use sp\_grantlogin instead for AD user

### Escape Tokens til at overføre SQL info til Powershell Jobbene

$BrugerID= [System.Security.Principal.WindowsIdentity]::GetCurrent().Name

$Maskine="$(ESCAPE\_SQUOTE(MACH))" # Pcnavn uden domain, uden instans

$Server="$(ESCAPE\_SQUOTE(SRVR))" # Pcnavn uden domain, med instans

$Inst="$(ESCAPE\_SQUOTE(INST))" # SQL instans

$MSSA="$(ESCAPE\_SQUOTE(MSSA))" # Master SQLServerAgent service name.

$Jobid="$(ESCAPE\_SQUOTE(JOBID))" # langt nummer

$SQLDIR="$(ESCAPE\_SQUOTE(SQLDIR))" # Sql Instal dir

$Stepid="$(ESCAPE\_SQUOTE(STEPID))" # Step nummer

$Start Date="$(ESCAPE\_SQUOTE(STRTDT))" # Job Start Dato

$Start Time="$(ESCAPE\_SQUOTE(STRTTM))" # Job Start kl.

$Date="$(ESCAPE\_SQUOTE(DATE))" # Dato Nu

$Time="$(ESCAPE\_SQUOTE(TIME))" # Klokken Nu

(<https://docs.microsoft.com/en-us/sql/ssms/agent/use-tokens-in-job-steps?view=sql-server-2017>)

## Klargør tom table, med same kolonner som den oprindelige.

select Top 0 \*

into #mvvtemp

from [dbo].[Version]

## Tøm / Truncate /slet indhold i Database

USE <DatabaseNavn>

EXEC sp\_MSforeachtable 'TRUNCATE TABLE ?'

## Kopier table til temp table

-- Drop #mvvtemp

IF OBJECT\_ID('tempdb..#mvvtemp') IS NOT NULL

DROP TABLE #mvvtemp

-- Kopier nuværende tabel

select \*

into #mvvtemp

from [dbo].[Version]

--Kontroller

select \* from #mvvtemp

Ret størrelse / type på kolonne I table (Tabel droppes & creates)

ALTER TABLE Version

ALTER COLUMN SQL\_CU NVARCHAR(10)

## List kolonne pr. Tabel pr. Database Søg CPR

--Oprette temp tabel # ColTblDb (sletter evt temp table først)

--Lister database og for hver db indsættes dbnavn, tabelnavn & kolonnenavn I temp tabel

IF OBJECT\_ID('tempdb..#ColTblDb') IS NOT NULL DROP TABLE #ColTblDb

Create table #ColTblDb (

DBname Varchar (100),

TaName Varchar (100),

CoName Varchar (100),

CoSize int

)

declare @str varchar(max) = ''

;with dbs as (select \* from sys.databases

where name not in ('master', 'tempdb', 'model', 'msdb','ReportServer','ReportServerTempDB')

and state = 0 ) -- (online)

select @str = @str + 'Insert into #ColTblDb select ''' + dbs.name + ''' as DBname, ta.name as TaName, co.name as CoName, co.max\_length as CoSize from ' +

dbs.name + '.sys.tables ta inner join ' +

dbs.name + '.sys.columns co ON co.object\_id = ta.object\_id; '

from dbs

exec(@str)

### -- Hent alle linier hvor der er fundet ”cpr” pr. tabel eller kolonne

select distinct DBname

,case when TaName like '%cpr%' then TaName else CoName end as CprValue

,case when TaName like '%cpr%' then 'Tabel' else 'Kolonne' end as CprSel

,case when TaName like '%cpr%' then Null else CoSize end as CoSize

from #ColTblDb where TaName like '%cpr%' or CoName like '%cpr%'

order by DBname

## -- Tæller forekomster af ordet ”cpr” i kolonnenavne & Tabelnavne pr. database

select DBname

,Sum(case when TaName like '%cpr%' then 1 else 0 end) as Cpr\_i\_Tabel

,Sum(case when CoName like '%cpr%' then 1 else 0 end) as Cpr\_i\_Kolonne

from #ColTblDb

where TaName like '%cpr%' or CoName like '%cpr%'

group by DBname

order by Sum(2+3) desc

## -- viser alle tabeller pr. database

select distinct DBname, TaName from #ColTblDb

Order by DBname, TaName

## -- viser antal tabeller pr. database

select DBname, Count(Distinct TaName) as antTab from #ColTblDb

group by DBname Order by DBname

## -- Show column pr. Table pr. Schema in Database relations

select c.name as column\_name ,

--distinct --if You use distinct instead of c.name You can list table / schema relations.

t.name as table\_name ,

s.name as schema\_name

,u.name AS Schema\_Owner

from sys.columns c

inner join sys.tables t on c.object\_id=t.object\_id

INNER JOIN sys.schemas AS s ON t.[schema\_id] = s.[schema\_id]

INNER JOIN sys.sysusers u ON u.uid = s.principal\_id

## --Schema Owner

/\* Alter Schema owner1 to original owner \*/

declare @sql nvarchar(max)

set @sql = ''

SELECT @sql = @sql+

'

print ''Transferring schema '+name+' to '+name+'''

ALTER AUTHORIZATION ON SCHEMA::'+name+' TO '+name+';

'

FROM sys.schemas

WHERE name IN (

SELECT s.Name --as SchemaName, u.name as ownerName

FROM sys.schemas s INNER JOIN sys.sysusers u

ON u.uid = s.principal\_id

where s.name not in('guest','INFORMATION\_SCHEMA','sys','public')

and s.name like 'db\_%'

and u.name NOT like s.name

)

ORDER BY name

Print @sql

execute (@sql)

## Alter Schema owner to dbo

/\* Alter Schema owner1 to dbo \*/

declare @sql1 nvarchar(max)

set @sql1 = ''

SELECT @sql1 = @sql1+

'

print ''Transfering schema '+name+' to dbo''

ALTER AUTHORIZATION ON SCHEMA::['+name+'] TO dbo;

'

FROM sys.schemas ss

WHERE name IN (

SELECT s.Name --as SchemaName, u.name as ownerName

FROM sys.schemas s

INNER JOIN sys.sysusers u ON u.uid = s.principal\_id

where s.name not in('guest','INFORMATION\_SCHEMA','sys','public')

and u.name not like 'db\_%'

and u.name <> 'dbo' --and u.name not ='dbo'

)

ORDER BY name

Print @sql1

execute (@sql1)

## Drop all normal users

/\* Drop all normal users \*/

declare @sql2 nvarchar(max)

set @sql2 = ''

SELECT @sql2 = @sql2+

'

print ''Dropping '+name+'''

DROP USER ['+name+']

'

FROM

sys.database\_principals dp

WHERE name NOT IN('dbo','guest','INFORMATION\_SCHEMA','sys','public')

AND type <> 'R'

AND NOT EXISTS

(

SELECT 1

FROM sys.schemas s

WHERE s.principal\_id = dp.principal\_id

)

ORDER BY name

Print @sql2

execute (@sql2)

## List alle db-users på instans vælg user via Temp table

IF OBJECT\_ID('tempdb..#User', 'U') IS NOT NULL

Begin

DROP TABLE #User;

End;

Create Table #User (DBname nvarchar (128) NOT NULL, DBUserName nvarchar (128) )

Truncate Table #User

Declare @komd as nvarchar(max)

Set @komd='

Use [?]

Print ''?''

Insert into #User

select ''?'',name from dbo.sysusers

Where (name not like ''%svd\_sqla%'' and name not like ''%svd\_sqle%'' and name not like ''%db\_%'')

and name not in (''sa'',''dbo'',''sys'',''public'',''guest'',''##MS\_SQLResourceSigningCertificate##'',''##MS\_SQLReplicationSigningCertificate##'',

''##MS\_SQLAuthenticatorCertificate##'',''##MS\_PolicySigningCertificate##'',''##MS\_SmoExtendedSigningCertificate##'',

''##MS\_PolicyTsqlExecutionLogin##'',''INFORMATION\_SCHEMA'',''DKSUND\L-ORG-MSSQL-Sysadmin'',

''NT SERVICE\SQLTELEMETRY'',''NT SERVICE\MSSQLSERVER'',''NT SERVICE\SQLSERVERAGENT'',''NT AUTHORITY\SYSTEM'',''NT AUTHORITY\LOCAL SERVICE'',

''NT SERVICE\SQLWriter'',''NT SERVICE\Winmgmt'',''SQLDBAcoll'',''SSI\MSSQLServerAdms'',''DKSUND\u-org-mssql-dba'',''SSI\svb-mssqlcoll'',

''##MS\_PolicyEventProcessingLogin##'',''##MS\_AgentSigningCertificate##'',''##MS\_SQLEnableSystemAssemblyLoadingUser##'',

''##MS\_SSISServerCleanupJobLogin##'')

'

Exec sp\_MSforeachdb @komd;

--Print @komd

Select \* from #User

where DBUserName like '%check%'

## Stop Running Processes

/\*To see the longest running transaction on your SQL Server instance, run the following statement.\*/

DBCC OPENTRAN

/\* If there are open transactions, DBCC OPENTRAN will provide a session\_id (SPID) of the connection that

has the transaction open.  You can pass this session\_id to sp\_who2 to determine which user has the connection open.

\*/

EXECUTE sp\_who2 spid

/\* Alternatively, you can run the following query to determine the user.\*/

SELECT \* FROM sys.dm\_exec\_sessions

WHERE session\_id = spid  —from DBCC OPENTRAN

/\* You can then kill the process blocking the greater good \*/

kill <SPID>

## Show Connections to db

Select s.host\_name,s.login\_name,s.original\_login\_name,s.status,s.login\_time from sys.dm\_exec\_sessions s

WHERE database\_id = db\_id('Nilex')

order by s.login\_name

## Kill Connections to db

USE [master];

DECLARE @kill varchar(8000) = '';

SELECT @kill = @kill + 'kill ' + CONVERT(varchar(5), session\_id) + ';'

FROM sys.dm\_exec\_sessions

WHERE database\_id = db\_id('Nilex')

Print(@kill);

EXEC(@kill);

## Take Offline / Online / Detache DB

USE master

GO

ALTER DATABASE nilex

SET OFFLINE WITH ROLLBACK IMMEDIATE

GO

-- vent nogle minutter til alle er timet ud.

ALTER DATABASE nilex

SET OnLINE WITH ROLLBACK IMMEDIATE

GO

-- man kan evt omdøbe db så der ikke er andre der får adgang.

-- Eller Detache

USE master

GO

EXEC sp\_detach\_db @dbname = nilex

Go

/\* You can determine the SQL statement being executed inside the transactions a couple of different ways.

First, you can use the DBCC INPUTBUFFER() statement to return the first part of the SQL statement\*/

DBCC INPUTBUFFER(spid)  —from DBCC OPENTRAN

--[?01-?09-?2017 13:03] Arne Johansen:

SELECT db\_name(es.database\_id) db, es.login\_name, es.login\_time, es.original\_login\_name, es.last\_request\_start\_time, ec.\* -- \* --ec.client\_net\_address, es.[program\_name], es.[host\_name], es.login\_name, COUNT(ec.session\_id) AS [connection count]

FROM sys.dm\_exec\_sessions AS es WITH (NOLOCK)

INNER JOIN sys.dm\_exec\_connections AS ec WITH (NOLOCK)

ON es.session\_id = ec.session\_id

where 1 like 'ssis%'

--Order by 1

/\* Delete all users in database \*/

declare @sql nvarchar(max)

set @sql = ''

SELECT **@sql** = @sql+

'

print ''Dropping '+name+'''

execute master.dbo.sp\_revokedbaccess '''+name+'''

'

FROM

dbo.sysusers

WHERE

name NOT IN('dbo','guest','INFORMATION\_SCHEMA','sys','public')

AND LEFT(name,3) <> 'db\_'

order by

name

Print @sql

execute ( @sql )

## CheckDB

-- Check konsistensen af en database

USE <db>

GO

DBCC CHECKDB -- WITH PHYSICAL\_ONLY; --fuld version eller light version ☺

GO

-- Check konsistensen af en tabel

dbcc checktable ('LMS.DS\_EkoKur')

Understanding DBCC Error Messages

After the DBCC CHECKDB command finishes, a message is written to the SQL Server error log. If the DBCC command successfully executes, the message indicates success and the amount of time that the command ran. If the DBCC command stops before completing the check because of an error, the message indicates that the command was terminated, a state value, and the amount of time the command ran. The following table lists and describes the state values that can be included in the message.

Error Reporting

A dump file (SQLDUMPnnnn.txt) is created in the SQL Server LOG directory whenever DBCC CHECKDB detects a corruption error. When the Feature Usage data collection and Error Reporting features are enabled for the instance of SQL Server, the file is automatically forwarded to Microsoft. The collected data is used to improve SQL Server functionality.

The dump file contains the results of the DBCC CHECKDB command and additional diagnostic output. Access is limited to the SQL Server service account and members of the sysadmin role. By default, the sysadmin role contains all members of the Windows BUILTIN\Administrators group and the local administrator's group. The DBCC command does not fail if the data collection process fails.

You could also put something like the following in place to read through the error log and send the output as an email.

CREATE TABLE #Errors (LogDate datetime, ProcessInfo varchar(32), MsgText varchar(max))

INSERT #Errors EXEC xp\_readerrorlog

SELECT \* FROM #Errors WHERE MsgText LIKE '%DBCC CHECKDB%'

ORDER BY LogDate

DROP TABLE #Errors

Please note that if you do a log switch, which you periodically should, this could miss some of the output so whatever solution you put in place you need to make it robust and test to verify that first of all you are doing CHECKDB on all your databases and secondly that you have a process in place for reviewing the output.

## T-SQL læs default data location fra Registry

DECLARE @returnValue NVARCHAR(500)

EXEC master..xp\_instance\_regread

@rootkey = N'HKEY\_LOCAL\_MACHINE',

@key = N'SOFTWARE\Microsoft\MSSQLServer\MSSQLServer',

@value\_name = N'DefaultData',

@value = @returnValue output

--PRINT @returnValue

Select @@SERVERNAME as SQLInstans, @returnValue as Defaultdata

## T-SQL Skriv default data location til Registry

USE [master]

GO

— Change default location for data files

EXEC   xp\_instance\_regwrite

       N'HKEY\_LOCAL\_MACHINE',

       N'Software\Microsoft\MSSQLServer\MSSQLServer',

       N'DefaultData',

       REG\_SZ,

       N'C:\MSSQL\Data'

GO

— Change default location for log files

EXEC   xp\_instance\_regwrite

       N'HKEY\_LOCAL\_MACHINE',

       N'Software\Microsoft\MSSQLServer\MSSQLServer',

       N'DefaultLog',

       REG\_SZ,

       N'C:\MSSQL\Logs'

GO

— Change default location for backups

EXEC   xp\_instance\_regwrite

       N'HKEY\_LOCAL\_MACHINE',

       N'Software\Microsoft\MSSQLServer\MSSQLServer',

       N'BackupDirectory',

       REG\_SZ,

       N'C:\MSSQL\Backups'

GO

## Error log location

SELECT \* FROM sys.dm\_os\_server\_diagnostics\_log\_configurations

## File Size

Select name, sum(size) Size from sys.master\_files

Where type\_desc='ROWS'

group by name

order by 2 desc

## Check size / størrelse Tempdb:

Højreklik på TempDB databasen, Åbn properties og check filer.

Dette viser dog kun sidst configurerede / start up størrelse location af TempDB.

Så for at få den aktuelle størrelse / placering af TempDB, kan vi køre

Exec sp\_helpdb 'TempDB';

## Oprydning i gamle Sletbare Tempdb filer

For at se alle fil definitioner af tempdb:

(Viser også tidligere tempdb definitioner, hvor fysiske filer er slettet / flyttet.)

Use master;

Select name, file\_id, ((size\*8)/1024) as SIZE\_IN\_MB, ((growth\*8)/1024) as Growth\_IN\_MB, type, Type\_desc, Physical\_Name From sys.master\_files m Where database\_id=2 Order By type, file\_id

Her kan vi sammenligne med sp\_helpdb resultatet, og finde filer der ikke er aktuelle mere, og fjerne fra Tempdb definitionerne:

ALTER DATABASE [tempdb] REMOVE FILE temp2;

ALTER DATABASE [tempdb] REMOVE FILE temp3;

ALTER DATABASE [tempdb] REMOVE FILE temp4;

Hvorefter det vil passe når man checker sys.master\_files

Hvis man skal klargøre til sletning af tempdb fil, skal man sikre at man har en tom tempdb fil:

USE [tempdb]

GO

DBCC SHRINKFILE (N'temp02', EMPTYFILE)

GO

ALTER DATABASE [tempdb] REMOVE FILE [temp02]

GO

### Andre check af Tempdb filer.

You can also use below query to find the actual current size of the TempDB.

select Name,((size\*8)/1024) as SIZE\_IN\_MB from Tempdb.sys.database\_files;

SELECT name AS FileName,

size\*1.0/128 AS FileSizeInMB,

CASE max\_size

WHEN 0 THEN 'Autogrowth is off.'

WHEN -1 THEN 'Autogrowth is on.'

ELSE 'Log file grows to a maximum size of 2 TB.'

END,

growth AS 'GrowthValue',

'GrowthIncrement' =

CASE

WHEN growth = 0 THEN 'Size is fixed.'

WHEN growth > 0 AND is\_percent\_growth = 0

THEN 'Growth value is in 8-KB pages.'

ELSE 'Growth value is a percentage.'

END

FROM tempdb.sys.database\_files;

GO

-- Determining the amount of free space in tempdb

SELECT SUM(unallocated\_extent\_page\_count) AS [free pages],

(SUM(unallocated\_extent\_page\_count)\*1.0/128) AS [free space in MB]

FROM sys.dm\_db\_file\_space\_usage;

-- Determining the amount of space used by the version store

SELECT SUM(version\_store\_reserved\_page\_count) AS [version store pages used],

(SUM(version\_store\_reserved\_page\_count)\*1.0/128) AS [version store space in MB]

FROM sys.dm\_db\_file\_space\_usage;

-- Determining the amount of space used by internal objects

SELECT SUM(internal\_object\_reserved\_page\_count) AS [internal object pages used],

(SUM(internal\_object\_reserved\_page\_count)\*1.0/128) AS [internal object space in MB]

FROM sys.dm\_db\_file\_space\_usage;

-- Determining the amount of space used by user objects

SELECT SUM(user\_object\_reserved\_page\_count) AS [user object pages used],

(SUM(user\_object\_reserved\_page\_count)\*1.0/128) AS [user object space in MB]

FROM sys.dm\_db\_file\_space\_usage;

-- Obtaining the space consumed by internal objects in all currently running tasks in each session

SELECT session\_id,

SUM(internal\_objects\_alloc\_page\_count) AS task\_internal\_objects\_alloc\_page\_count,

SUM(internal\_objects\_dealloc\_page\_count) AS task\_internal\_objects\_dealloc\_page\_count

FROM sys.dm\_db\_task\_space\_usage

GROUP BY session\_id;

-- Obtaining the space consumed by internal objects in the current session for both running and completed tasks

SELECT R2.session\_id,

R1.internal\_objects\_alloc\_page\_count

+ SUM(R2.internal\_objects\_alloc\_page\_count) AS session\_internal\_objects\_alloc\_page\_count,

R1.internal\_objects\_dealloc\_page\_count

+ SUM(R2.internal\_objects\_dealloc\_page\_count) AS session\_internal\_objects\_dealloc\_page\_count

FROM sys.dm\_db\_session\_space\_usage AS R1

INNER JOIN sys.dm\_db\_task\_space\_usage AS R2 ON R1.session\_id = R2.session\_id

GROUP BY R2.session\_id, R1.internal\_objects\_alloc\_page\_count,

R1.internal\_objects\_dealloc\_page\_count;;

[tempdb database - SQL Server | Microsoft Docs](https://docs.microsoft.com/en-us/sql/relational-databases/databases/tempdb-database?view=sql-server-ver15)

[Recommendations to reduce allocation contention - SQL Server | Microsoft Docs](https://docs.microsoft.com/en-US/troubleshoot/sql/performance/recommendations-reduce-allocation-contention)

## Move Tempdb to new disk.

1. You need to create the G:\NewTempDB\ directory\folder (or whatever name you choose to move the files too).
2. After you created NewTempDB folder, then you need to right click -> properties -> security and add your SQL Service account, set full control permissions for it.
3. Now you can run the alter database command, make sure that the files end up in the created folder in the command.
4. ALTER DATABASE tempdb MODIFY FILE (name = tempdb, filename='G:\NewTempDB\tempdb.mdf');
5. ALTER DATABASE tempdb MODIFY FILE (name = templog, filename='G:\NewTempDB\templog.ldf');
6. Restart the SQL Service
7. Delete the old tempdb files on the C:\ drive

Dette kan gøres ved: (med tak til Brent Ozar)

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

+ ' FILENAME = ''Q:\System\MSSQL15.MSSQLSERVER\MSSQL\Data\' + f.name

+ CASE WHEN f.type = 1 THEN '.ldf' WHEN f.file\_id > 2 THEN '.ndf' ELSE '.mdf' END

+ ''', SIZE = 1048576KB , MAXSIZE = 102400000KB , FILEGROWTH = 262144KB);'

FROM sys.master\_files f

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

Hvor man først retter lokationen, size, maxsize & Filegrowth

Så får man liste over tempdb filer på ny disk, tilføjer evt. tilføjelser:

ALTER DATABASE [tempdb] ADD FILE ( NAME = N'Tempdev6', FILENAME = N'Q:\System\MSSQL12.MSSQLSERVER\MSSQL\Data\Tempdev6.ndf' , SIZE = 1048576KB , MAXSIZE = 102400000KB , FILEGROWTH = 262144KB )

ALTER DATABASE [tempdb] ADD FILE ( NAME = N'Tempdev7', FILENAME = N'Q:\System\MSSQL12.MSSQLSERVER\MSSQL\Data\Tempdev7.ndf' , SIZE = 1048576KB , MAXSIZE = 102400000KB , FILEGROWTH = 262144KB )

Fjerner maxsize på f.eks. templog

Kører scriptet

Virker først efter genstart af tempdb

## Shrink Tempdb in place

Use tempdb

Go

sp\_spaceused @updateusage=true;

--beregn % =ønsket friplads\*100/total plads (f.eks. 25)

dbcc shrinkdatabase (tempdb, 25)

-- evt shrink enkeltfiler:

USE [tempdb]

GO

DBCC SHRINKFILE (N'temp8' , 90000)

GO

USE [master]

GO

ALTER DATABASE [tempdb] MODIFY FILE ( NAME = N'temp8', SIZE = 90022656KB )

GO

## Check Brug af TempDB når denne vokser meget.

SQL Server TempDB Usage Queries

The following query uses *sys. dm\_db\_session\_space\_usage* view to show the total and net allocation of both user and internal objects and the last query executed by the session. Notice that in order to get the space allocated in megabytes we need to divide the number of pages by 128. That’s because the page size is 8 kilobytes resulting in 128 pages per megabyte.

SELECT SS.session\_id , SS.database\_id ,

CAST(SS.user\_objects\_alloc\_page\_count / 128 AS DECIMAL(15, 2)) [Total Allocation User Objects MB] ,

CAST(( SS.user\_objects\_alloc\_page\_count

- SS.user\_objects\_dealloc\_page\_count ) / 128 AS DECIMAL(15, 2)) [Net Allocation User Objects MB] ,

CAST(SS.internal\_objects\_alloc\_page\_count / 128 AS DECIMAL(15, 2)) [Total Allocation Internal Objects MB] ,

CAST(( SS.internal\_objects\_alloc\_page\_count

- SS.internal\_objects\_dealloc\_page\_count ) / 128 AS DECIMAL(15,

2)) [Net Allocation Internal Objects MB] ,

CAST(( SS.user\_objects\_alloc\_page\_count

+ internal\_objects\_alloc\_page\_count ) / 128 AS DECIMAL(15, 2)) [Total Allocation MB] ,

CAST(( SS.user\_objects\_alloc\_page\_count

+ SS.internal\_objects\_alloc\_page\_count

- SS.internal\_objects\_dealloc\_page\_count

- SS.user\_objects\_dealloc\_page\_count ) / 128 AS DECIMAL(15, 2)) [Net Allocation MB] ,

T.text [Query Text]

FROM sys.dm\_db\_session\_space\_usage SS

LEFT JOIN sys.dm\_exec\_connections CN ON CN.session\_id = SS.session\_id

OUTER APPLY sys.dm\_exec\_sql\_text(CN.most\_recent\_sql\_handle) T

On the other hand, the next query uses sys.dm\_db\_task\_space\_usage view to show the total and net allocation of user and internal objects as well as the query being executed by each active task.

SELECT TS.session\_id ,

TS.request\_id ,

TS.database\_id ,

CAST(TS.user\_objects\_alloc\_page\_count / 128 AS DECIMAL(15, 2)) [Total Allocation User Objects MB] ,

CAST(( TS.user\_objects\_alloc\_page\_count

- TS.user\_objects\_dealloc\_page\_count ) / 128 AS DECIMAL(15, 2)) [Net Allocation User Objects MB] ,

CAST(TS.internal\_objects\_alloc\_page\_count / 128 AS DECIMAL(15, 2)) [Total Allocation Internal Objects MB] ,

CAST(( TS.internal\_objects\_alloc\_page\_count

- TS.internal\_objects\_dealloc\_page\_count ) / 128 AS DECIMAL(15,

2)) [Net Allocation Internal Objects MB] ,

CAST(( TS.user\_objects\_alloc\_page\_count

+ internal\_objects\_alloc\_page\_count ) / 128 AS DECIMAL(15, 2)) [Total Allocation MB] ,

CAST(( TS.user\_objects\_alloc\_page\_count

+ TS.internal\_objects\_alloc\_page\_count

- TS.internal\_objects\_dealloc\_page\_count

- TS.user\_objects\_dealloc\_page\_count ) / 128 AS DECIMAL(15, 2)) [Net Allocation MB] ,

T.text [Query Text]

FROM sys.dm\_db\_task\_space\_usage TS

INNER JOIN sys.dm\_exec\_requests ER ON ER.request\_id = TS.request\_id

AND ER.session\_id = TS.session\_id

OUTER APPLY sys.dm\_exec\_sql\_text(ER.sql\_handle) T

But it doesn’t end here, because we can merge both queries to get space usage by session including the current requests.

SELECT COALESCE(T1.session\_id, T2.session\_id) [session\_id] , T1.request\_id ,

COALESCE(T1.database\_id, T2.database\_id) [database\_id],

COALESCE(T1.[Total Allocation User Objects], 0)

+ T2.[Total Allocation User Objects] [Total Allocation User Objects] ,

COALESCE(T1.[Net Allocation User Objects], 0)

+ T2.[Net Allocation User Objects] [Net Allocation User Objects] ,

COALESCE(T1.[Total Allocation Internal Objects], 0)

+ T2.[Total Allocation Internal Objects] [Total Allocation Internal Objects] ,

COALESCE(T1.[Net Allocation Internal Objects], 0)

+ T2.[Net Allocation Internal Objects] [Net Allocation Internal Objects] ,

COALESCE(T1.[Total Allocation], 0) + T2.[Total Allocation] [Total Allocation] ,

COALESCE(T1.[Net Allocation], 0) + T2.[Net Allocation] [Net Allocation] ,

COALESCE(T1.[Query Text], T2.[Query Text]) [Query Text]

FROM ( SELECT TS.session\_id ,

TS.request\_id ,

TS.database\_id ,

CAST(TS.user\_objects\_alloc\_page\_count / 128 AS DECIMAL(15,

2)) [Total Allocation User Objects] ,

CAST(( TS.user\_objects\_alloc\_page\_count

- TS.user\_objects\_dealloc\_page\_count ) / 128 AS DECIMAL(15,

2)) [Net Allocation User Objects] ,

CAST(TS.internal\_objects\_alloc\_page\_count / 128 AS DECIMAL(15,

2)) [Total Allocation Internal Objects] ,

CAST(( TS.internal\_objects\_alloc\_page\_count

- TS.internal\_objects\_dealloc\_page\_count ) / 128 AS DECIMAL(15,

2)) [Net Allocation Internal Objects] ,

CAST(( TS.user\_objects\_alloc\_page\_count

+ internal\_objects\_alloc\_page\_count ) / 128 AS DECIMAL(15,

2)) [Total Allocation] ,

CAST(( TS.user\_objects\_alloc\_page\_count

+ TS.internal\_objects\_alloc\_page\_count

- TS.internal\_objects\_dealloc\_page\_count

- TS.user\_objects\_dealloc\_page\_count ) / 128 AS DECIMAL(15,

2)) [Net Allocation] ,

T.text [Query Text]

FROM sys.dm\_db\_task\_space\_usage TS

INNER JOIN sys.dm\_exec\_requests ER ON ER.request\_id = TS.request\_id

AND ER.session\_id = TS.session\_id

OUTER APPLY sys.dm\_exec\_sql\_text(ER.sql\_handle) T

) T1

RIGHT JOIN ( SELECT SS.session\_id ,

SS.database\_id ,

CAST(SS.user\_objects\_alloc\_page\_count / 128 AS DECIMAL(15,

2)) [Total Allocation User Objects] ,

CAST(( SS.user\_objects\_alloc\_page\_count

- SS.user\_objects\_dealloc\_page\_count )

/ 128 AS DECIMAL(15, 2)) [Net Allocation User Objects] ,

CAST(SS.internal\_objects\_alloc\_page\_count / 128 AS DECIMAL(15,

2)) [Total Allocation Internal Objects] ,

CAST(( SS.internal\_objects\_alloc\_page\_count

- SS.internal\_objects\_dealloc\_page\_count )

/ 128 AS DECIMAL(15, 2)) [Net Allocation Internal Objects] ,

CAST(( SS.user\_objects\_alloc\_page\_count

+ internal\_objects\_alloc\_page\_count ) / 128 AS DECIMAL(15,

2)) [Total Allocation] ,

CAST(( SS.user\_objects\_alloc\_page\_count

+ SS.internal\_objects\_alloc\_page\_count

- SS.internal\_objects\_dealloc\_page\_count

- SS.user\_objects\_dealloc\_page\_count )

/ 128 AS DECIMAL(15, 2)) [Net Allocation] ,

T.text [Query Text]

FROM sys.dm\_db\_session\_space\_usage SS

LEFT JOIN sys.dm\_exec\_connections CN ON CN.session\_id = SS.session\_id

OUTER APPLY sys.dm\_exec\_sql\_text(CN.most\_recent\_sql\_handle) T

) T2 ON T1.session\_id = T2.session\_id

##### Next Steps

* In the following tip you can see how to troubleshoot a sort spilling to TempDB: [Correct SQL Server TempDB Spills in Query Plans Caused by Outdated Statistics](https://www.mssqltips.com/sqlservertip/4132/correct-sql-server-tempdb-spills-in-query-plans-caused-by-outdated-statistics/).
* It’s very important to monitor TempDB growth, the following tip will teach you how to implement an alert when TempDB is growing: [SQL Server Alert for TempDB Growing Out of Control](https://www.mssqltips.com/sqlservertip/3276/sql-server-alert-for-tempdb-growing-out-of-control/).
* In case you need to track Buffer Pool space usage the following tip will guide you: [Determine objects consuming the largest amount of space in the SQL Server buffer pool](https://www.mssqltips.com/sqlservertip/2239/determine-objects-consuming-the-largest-amount-of-space-in-the-sql-server-buffer-pool/)
* If you also need to [Determine SQL Server memory use by database and object](https://www.mssqltips.com/sqlservertip/2393/determine-sql-server-memory-use-by-database-and-object/), this tip is the one for you.
* You don’t know what Dynamic Management Views are? Don’t worry; this tip explain this to you: [Dynamic Management Views](https://www.mssqltips.com/sqlservertutorial/273/dynamic-management-views/).
* Check out the [SQL Server Dynamic Management Views and Functions Tip](https://www.mssqltips.com/sql-server-tip-category/31/dynamic-management-views-and-functions/) category.

## Move Database Files / Flyt DB filer

--[2021-06-07,STGN] data + log filer flyttet .

-- <https://www.mssqltips.com/sqlservertip/1774/move-sql-server-transaction-log-files-to-a-different-location-via-tsql-and-ssms/>

-- Nuværende source er ligegyldig.

-- Ønsket destniation

-- 'R:\System\MSSQL15.MSSQLSERVER\MSSQL\Data\IB\_KID.mdf',

-- 'S:\System\MSSQL15.MSSQLSERVER\MSSQL\Log\IB\_KID\_log.ldf';

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

--- s-epidb02-t.dksund.dk

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

Use MASTER

**GO**

-- Set database to mode (Restricted\_user/db\_owner)

ALTER DATABASE [IB\_KID] SET Restricted\_user WITH ROLLBACK IMMEDIATE

**GO**

-- Detach the database

sp\_detach\_db 'IB\_KID'

**GO**

----------------- Go fix OS samt fil placering -

USE master

**GO**

-- Now Attach the database

sp\_attach\_DB 'IB\_KID',

'R:\System\MSSQL15.MSSQLSERVER\MSSQL\Data\IB\_KID.mdf',

'S:\System\MSSQL15.MSSQLSERVER\MSSQL\Log\IB\_KID\_log.ldf';

**GO**

ALTER DATABASE [IB\_KID] SET multi\_user WITH ROLLBACK IMMEDIATE

**GO**

## SHRINK DATABASE / FILE

Problem: SQLLog(S: ) er ved at løbe fuld. (disk plads)

Problem: DB SQL\_Logfil (T- log) løber løbsk, og ikke kan blive større, da disken er ved at være fyldt

Så skal der ryddes op i logfil. Dette forekommer når Databasen er sat til Full recovery.

Eller hvis et job laver RIGTIGT mange rettelser i en database uden at committe disse.

Når man tager en Fuld backup af en database har man en kopi af databasen.

Hvis databasen er i brug mens der tages backup, kan der ske ændringer under backuppen, der ikke kommer med i backuppen.

Databasens Transaktions-log indeholder de database transaktioner, der er foretaget siden sidste T-log backup (i klumper).

Når man tager en Transaktions Log Backup, har man backup af de ændringer der er sket med databasen siden starten af sidste backup (log / fuld).

Om man kører 10 t-log backup, eller 1 om dagen er ligegyldigt, det fylder det samme på backup mediet.

Forskellen er at hver gang man kører t-log backup, tømmer man t-loggen (i klumper), hvorved man sørger for at den ikke vokser uhæmmet.

Der bliver dog IKKE ændret på størrelsen af Log-filen ved en backup, den ligger der som en stor (næsten tom) placeholder på disken.

Da log filerne kun kan formindskes ved at tømme dem (T-Log backup) efterfulgt af DBCC SHRINKFILE, skal man sørge for at t-log backup kører regelmæssigt. (hver time)

Man skal sikre at der har været en vellykket Fuld backup (for nyligt), og at efterfølgende log-backup er gået godt.

Hyppigheden af t-log backup definerer hvor meget systemet kan tåle at miste.

#### Simple

Hvis databasen er sat til simple recovery, kan man checke logforbrug (se nedenfor), checke sidste full backup, (evt. køre en ny) og shrinke (se nedenfor)

Man kan checke sidste log-backup ved:

USE MSDB;

Select @@SERVERNAME as SqlInstance

,B.database\_name

,type

,max(B.backup\_finish\_date) as backup\_finish\_date

,MAX(D.recovery\_model\_desc) as recovery\_model\_desc

,MAX(D.log\_reuse\_wait\_desc) as log\_reuse\_wait\_desc

from backupset B

JOIN sys.databases D

ON D.name = B.database\_name

Where B.type='L' --her kan du bare ændre til fx D (D = database/full backup)

Group by B.database\_name, type

Order by B.database\_name

Hver gang man kører t-log backup, tømmer man t-loggen, hvorved man sørger for at den ikke vokser uhæmmet.

(BS) For at BrandSlukke kan man definere en ekstra t-log på en anden disk, der så bør ryddes op efter nedenstående shrink.

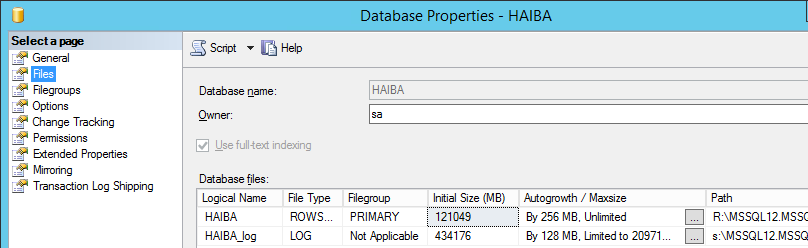
(BS) Hvis der er brug for den pågældende log-størrelse, så bør log disken udvides. (nemt hvis virtuel disk.)

Check windirstat, der nok vil pege på log-folderen (S:\MSSQL12.MSSQLSERVER\_TEST\MSSQL\Data)

Her kan man sortere på data størrelse, og koncentrere sig om de største.

I SSMS højreklikkes på database navnet, og der vælges properties.

Hvis logfilen en væsentligt større end den tilsvarende datafil:



Kan man undersøge:

/\* Vis log file size og %brugt \*/

use master

DBCC SQLPERF (LOGSPACE)



Her ser vi (på log space used %at Loggen er fyldt af tomme områder der skal reorganiseres.

/\* find navn på T-Log fil \*/

use HAIBA

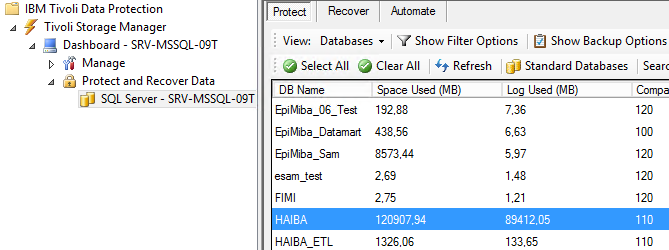
SELECT name

FROM sys.database\_files

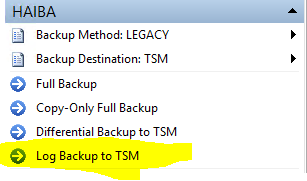
WHERE type\_desc = 'LOG'

Start ”DP for SQL Management Console”

Gå til rod og vælg database



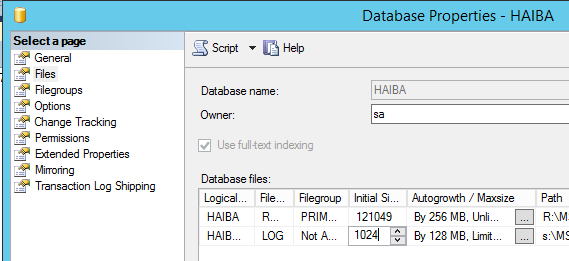
Vælg Log backup til TSM



Afvent afsluttet backup



Ret Properties\Files\log file størrelse



Husk maxsize unlimited og værdier i Mb

/\* Skrink log filen \*/

DBCC SHRINKFILE (HAIBA\_log ,1024)

Go

Kør t-log backup, ret størrelse og shrink, nogle gange, indtil t-log har en passende størrelse.

## Vis VLF (virtual log files) for en db.

/\* vis vlf (virtual log files) for en db. \*/

DBCC LOGINFO () WITH NO\_INFOMSGS;

Go

/\*Størrelse kan checkes ved: \*/

SELECT db.[name] AS [Database Name],

SUSER\_SNAME(db.owner\_sid) AS [Database Owner],

db.recovery\_model\_desc AS [Recovery Model],

--db.state\_desc, --db.containment\_desc,

--db.log\_reuse\_wait\_desc AS [Log Reuse Wait Description],

CONVERT(DECIMAL(18,2), ls.cntr\_value/1024.0) AS [Log Size (MB)],

CONVERT(DECIMAL(18,2), ls.cntr\_value/1024.0/1024.0) AS [Log Size (GB)],

CONVERT(DECIMAL(18,2), lu.cntr\_value/1024.0) AS [Log Used (MB)],

CONVERT(DECIMAL(18,2), lu.cntr\_value/1024.0/1024.0) AS [Log Used (GB)],

CAST(CAST(lu.cntr\_value AS FLOAT) / CAST(ls.cntr\_value AS FLOAT)AS DECIMAL(18,2)) \* 100 AS [Log Used %],

--db.[compatibility\_level] AS [DB Compatibility Level],

db.page\_verify\_option\_desc AS [Page Verify Option]

--db.is\_auto\_create\_stats\_on, db.is\_auto\_update\_stats\_on, --db.is\_auto\_update\_stats\_async\_on, --db.is\_parameterization\_forced,

--db.snapshot\_isolation\_state\_desc, --db.is\_read\_committed\_snapshot\_on,

--db.is\_auto\_close\_on, db.is\_auto\_shrink\_on

--db.target\_recovery\_time\_in\_seconds,

--db.is\_cdc\_enabled, db.is\_published, db.is\_distributor, db.is\_encrypted,

--db.group\_database\_id, db.replica\_id,db.is\_memory\_optimized\_elevate\_to\_snapshot\_on,

--db.delayed\_durability\_desc, db.is\_auto\_create\_stats\_incremental\_on,

--db.is\_encrypted, de.encryption\_state, de.percent\_complete, de.key\_algorithm, de.key\_length

FROM sys.databases AS db WITH (NOLOCK)

INNER JOIN sys.dm\_os\_performance\_counters AS lu WITH (NOLOCK)

ON db.name = lu.instance\_name

INNER JOIN sys.dm\_os\_performance\_counters AS ls WITH (NOLOCK)

ON db.name = ls.instance\_name

LEFT OUTER JOIN sys.dm\_database\_encryption\_keys AS de WITH (NOLOCK)

ON db.database\_id = de.database\_id

WHERE lu.counter\_name LIKE N'Log File(s) Used Size (KB)%'

AND ls.counter\_name LIKE N'Log File(s) Size (KB)%'

AND ls.cntr\_value > 0

AND db.[name] like 'HAIBA%'

ORDER BY db.[name] OPTION (RECOMPILE);

Ret I SD 272174 hvis der er ændringer til t-log hyppighed.

/\* Hvis der ikke er installeret DP for sql (TSM) kør manuel til disk. \*/

Use D3\_SOP

--BACKUP DATABASE [D3\_SOP] TO DISK = N'C:\Backup\D3\_SOP\_Full' WITH NOFORMAT, NOINIT, NAME = N'D3\_SOP-Full Database Backup', SKIP, NOREWIND, NOUNLOAD, STATS = 10

--GO

--declare @backupSetId as int

--select @backupSetId = position from msdb..backupset where database\_name=N'D3\_SOP' and backup\_set\_id=(select max(backup\_set\_id) from msdb..backupset where database\_name=N'D3\_SOP' )

--if @backupSetId is null begin raiserror(N'Verify failed. Backup information for database ''D3\_SOP'' not found.', 16, 1) end

--RESTORE VERIFYONLY FROM DISK = N'C:\Backup\D3\_SOP\_Full' WITH FILE = @backupSetId, NOUNLOAD, NOREWIND

--GO

--BACKUP LOG [D3\_SOP] TO DISK = N'C:\Backup\D3\_SOP\_Log' WITH NOFORMAT, NOINIT, NAME = N'D3\_SOP-Transaction Log Backup', SKIP, NOREWIND, NOUNLOAD, STATS = 10

--GO

BACKUP LOG [D3\_SOP] TO DISK = N'C:\Backup\D3\_SOP\_Log1' WITH NOFORMAT, NOINIT, NAME = N'D3\_SOP-Transaction Log Backup', SKIP, NOREWIND, NOUNLOAD, STATS = 10

GO

## Ryd op I Aktive Connections til DB

### List aktive forbindelser:

CREATE TABLE #sp\_who2 (SPID INT,Status VARCHAR(255),

Login VARCHAR(255),HostName VARCHAR(255),

BlkBy VARCHAR(255),DBName VARCHAR(255),

Command VARCHAR(255),CPUTime INT,

DiskIO INT,LastBatch VARCHAR(255),

ProgramName VARCHAR(255),SPID2 INT,

REQUESTID INT)

INSERT INTO #sp\_who2 EXEC sp\_who2

SELECT \*

FROM #sp\_who2

-- Add any filtering of the results here :

WHERE DBName not in ('master','msdb','model','tempdb','dba\_db')

and Status <> 'sleeping'

-- Add any sorting of the results here :

ORDER BY DBName ASC

DROP TABLE #sp\_who2

### List active SQL kald

SELECT r.start\_time [Start Time],session\_ID [SPID],

DB\_NAME(database\_id) [Database],

SUBSTRING(t.text,(r.statement\_start\_offset/2)+1,

CASE WHEN statement\_end\_offset=-1 OR statement\_end\_offset=0

THEN (DATALENGTH(t.Text)-r.statement\_start\_offset/2)+1

ELSE (r.statement\_end\_offset-r.statement\_start\_offset)/2+1

END) [Executing SQL],

Status,command,wait\_type,wait\_time,wait\_resource,

last\_wait\_type

FROM sys.dm\_exec\_requests r

OUTER APPLY sys.dm\_exec\_sql\_text(sql\_handle) t

WHERE session\_id != @@SPID -- don't show this query

AND session\_id > 50 -- don't show system queries

AND DB\_NAME(database\_id) not in ('master','msdb','model','tempdb','DBA\_DB')

ORDER BY r.start\_time

### Kan ryddes ved at tage offline, og derefter online igen, men husk ”with rollback immediate”

-- Take the Database Offline  --- DebuggerStepThroughAttribute all users off

ALTER DATABASE[STPS\_KontaktOpsporing] SET OFFLINE WITH ROLLBACK IMMEDIATE

**GO**

--Take the Database Online

ALTER DATABASE[STPS\_KontaktOpsporing] SET ONLINE

**GO**

### Men endnu bedre begræns til sysadmins

ALTER DATABASE[STPS\_KontaktOpsporing] SET RESTRICTED\_USER  WITH ROLLBACK IMMEDIATE

**GO**

ALTER DATABASE[STPS\_KontaktOpsporing] SET MULTI\_URSERS

**GO**

## Opret en SQL linked Server

En Linked Server I MS SQL er en server der er forhåndsgodkendt og forbundet, så man kan bruge en eller flere databaser som om de var lokale.

Dette virker dog kun direkte fra serveren til en anden server, det er ikke muligt at tilgå serveren fra en tredje server/client og bruge linked server mellem disse. Der kræves en slags Pass-through Authentication og har problem med Double Hop privilegier, se f.eks. nedenstående:

[SQL Server: Curse Of Linked Server Security And The Fix: Pass-through Authentication – SQL Jana (wordpress.com)](https://sqljana.wordpress.com/2017/06/16/sql-server-curse-of-linked-server-security-and-the-fix-pass-through-authentication/)

[Microsoft Made an Easy Button for SPN and Double Hop Issues | HoB (houseofbrick.com)](https://houseofbrick.com/microsoft-made-an-easy-button-for-spn-and-double-hop-issues/)

Man oprettet en linked server lettest via TSQL via sp\_addlinkedserver

og Sikkerhedsforbindelsen via sp\_addlinkedsrvlogin kommandoerne.

Man kan forbinde med egen bruger (man skal have privilegier begge steder)

Eller servicekonto der har privilegier på destinationen (password i fri tekst??)

### For at linke en hel SQL server og forbinde via egen windows konto:

USE [master]

GO

EXEC master.dbo.sp\_addlinkedserver @server = N'S-MSSQLASIS04-P', @srvproduct=N'', @provider=N'SQLOLEDB'

EXEC master.dbo.sp\_addlinkedsrvlogin @rmtsrvname = N'S-MSSQLASIS04-P', @locallogin = NULL , @useself = N'True'

GO

SELECT name FROM [S-MSSQLASIS04-P].master.sys.databases

–Eller mere specifikt

SELECT DBname FROM [S-MSSQLASIS04-P].DBA\_DB.dbo.DB;

Her får man en Linked Server ved navn [S-MSSQLASIS04-P]

Og man bruger sine egne rettigheder på den forbundne server hverken mere eller mindre.

Man kan forbinde med anden bruger i stedet for egen bruger.

### Det kan være en fordel at oprette en Linked Server til en enkelt Database

USE [master]

GO

declare @SQLS as nvarchar(50) = N's-mssqlasis04-p'

declare @SQLdb as nvarchar(50) = N'DBA\_DB'

declare @SQLta as nvarchar(50) = N'DB'

declare @SQLN as nvarchar(50) = N'AsisDb'

declare @SQLu as nvarchar(50) = N'SQLDBAcoll'

declare @SQLp as nvarchar(50) = '<pw>'

EXEC master.dbo.sp\_addlinkedserver @server = @SQLN, @srvproduct=@SQLN, @provider=N'SQLNCLI', @datasrc=@SQLS, @catalog=@SQLDB

EXEC master.dbo.sp\_addlinkedsrvlogin @rmtsrvname=@SQLN,@useself=N'False',@locallogin=NULL,@rmtuser=@SQLU,@rmtpassword=@SQLp

Her får man en Linked Server: AsisDb der kun peger på en database, men forbundet via en SQL konto.

### Så man opretter en Linked Server til en DB med egen windows konto:

USE [master]

GO

declare @SQLS as nvarchar(50) = N's-mssqlasis04-p'

declare @SQLdb as nvarchar(50) = N'DBA\_DB'

declare @SQLN as nvarchar(50) = N'AsisDba\_Db3'

EXEC master.dbo.sp\_addlinkedserver @server = @SQLN, @srvproduct=@SQLN, @provider=N'SQLNCLI', @datasrc=@SQLS, @catalog=@SQLDB

-- link med egen windows bruger

EXEC master.dbo.sp\_addlinkedsrvlogin @rmtsrvname = @SQLN, @locallogin = NULL , @useself = N'True', @rmtuser = N''

Skifter til SQL Connection Sikkerhed:

-- Use SQL USER Remote connect

declare @SQLu as nvarchar(50) = N'SQLDBAcoll'

declare @SQLp as nvarchar(50) = '<pw>'

EXEC master.dbo.sp\_addlinkedsrvlogin @rmtsrvname=@SQLN,@useself=N'False',@locallogin=NULL,@rmtuser=@SQLU,@rmtpassword=@SQLp

GO

Forbinder specifik AD-bruger til remote bruger

-- Link lokal AD-bruger som remote bruger

declare @SQLu as nvarchar(50) = N'dksund\adm\_mive'

EXEC master.dbo.sp\_addlinkedsrvlogin @rmtsrvname = N'AsisDba\_Db3', @locallogin = @SQLu, @useself = N'True'

Go

declare @SQLu as nvarchar(50) = N'dksund\adm\_mive'

declare @SQLp as nvarchar(50) = '<pw>'

EXEC master.dbo.sp\_addlinkedsrvlogin @rmtsrvname = N'AsisDba\_Db3', @locallogin = NULL , @useself = N'False', @rmtuser = @SQLu, @rmtpassword = @SQLp

GO

Og tilbage til egen bruger

USE [master]

GO

declare @SQLN as nvarchar(50) = N'AsisDba\_Db3'

-- link med egen windows bruger

EXEC master.dbo.sp\_addlinkedsrvlogin @rmtsrvname = @SQLN, @locallogin = NULL , @useself = N'True', @rmtuser = N''

### For at slette Linked Server Login:

USE [master]

GO

/\*\*\*\*\*\* Object: Login [] Script Date: 11-10-2021 16:41:41 \*\*\*\*\*\*/

EXEC master.dbo.sp\_droplinkedsrvlogin @rmtsrvname = N'AsisDba\_Db3', @locallogin = NULL

GO

### Og slette Linked Server

USE [master]

GO

/\*\*\*\*\*\* Object: LinkedServer [S-MSSQLASIS04-P] Script Date: 11-10-2021 17:51:51 \*\*\*\*\*\*/

EXEC master.dbo.sp\_dropserver @server=N'S-MSSQLASIS04-P', @droplogins='droplogins'

GO

## Rapport over installeret SQL

Installation Center can be launched from the Start Menu, under the SQL Server entry.

Click on the “Installed SQL Server features discovery report”

You can also generate the Discovery report through the command line. Run “Setup.exe /Action=RunDiscovery” from a command prompt

If you add “/q” to the command line above no UI will be shown, but the report will still be created in %ProgramFiles%\Microsoft SQL Server\100\Setup Bootstrap\Log\20091112\_082147.

## Tilføjelse af funktioner til Installeret SQL Server inkl. Tablediff.exe.

Start sql installations Center fra SQL Server installations .iso

Skift til cd

.\setup.exe

Installation

Klick på “New SQL Server stand-alone Installation or add features to existing installation”

Installation Type: Add features to existing instance (check version) Next

Her kan man installere SQL Server Replication inkl. Tablediff.exe der er et cmd – baseret værktøj der kan sammenligne tabeller / databaser med resultat: et script, der kan rette til efter forskellen der i mellem.

Gode anbefalinger & vejledninger på nettet. Men virker ”KUN” mod SQL Server.

/\*

Replication inkl. Tablediff.exe kan kopieres til anden server ved at kopiere folderen

C:\Program Files\Microsoft SQL Server\150\COM

For SQL Server 2019

[SQL Command Prompt Utilities (Database Engine) - SQL Server | Microsoft Docs](https://docs.microsoft.com/en-us/sql/tools/command-prompt-utility-reference-database-engine?view=sql-server-ver15)

[SQL Server tablediff utility - Simple Talk (red-gate.com)](https://www.red-gate.com/simple-talk/databases/sql-server/tools-sql-server/sql-server-tablediff-utility/)

\*/

Udover Replication, kan man også vælge Machine Learning med

* R (fra SQL Server 2016 sp1 ++)
* Python (fra SQL Server2019 ++)
* Java

Samt PolyBase, SSAS & SSIS

Efter installation husk at opgradere med SP / CU

PolyBase Links:

[Introducing data virtualization with PolyBase - SQL Server | Microsoft Docs](https://docs.microsoft.com/en-us/sql/relational-databases/polybase/polybase-guide?redirectedfrom=MSDN&view=sql-server-ver15)

[What Does Installing Polybase Add to SQL Server? - Microsoft Tech Community](https://techcommunity.microsoft.com/t5/core-infrastructure-and-security/what-does-installing-polybase-add-to-sql-server/ba-p/370939)

## Find Databaser & Disk fil navn / placering

Use Master

SELECT mf.database\_id

,DB\_NAME(database\_id) AS DatabaseName

,name AS LogicalFileName

,physical\_name AS PhysicalFileName

,state

,state\_desc

,cast(size as int)/128\*1024 as SizeKb

,max\_size

,size

,growth

FROM sys.master\_files AS mf

where mf.database\_id > 4

Man kan udvælge specific dbnavn (fra beregnet funktion) ved at have output fra ovenstående nested i ny select som f.eks.

Select \* from (SELECT database\_id,DB\_NAME(database\_id) DatabaseName,name LogicalFileName

,physical\_name PhysicalFileName,State,State\_desc ,cast(size as int)/128\*1024 as SizeKb ,Size,Max\_size,Growth

FROM sys.master\_files AS mf) as q

where q.DatabaseName Like 'di%' order by q.size

## Database (suspect) eller (recovery Pending)

<https://www.sqlskills.com/blogs/paul/creating-detaching-re-attaching-and-fixing-a-suspect-database/>

<https://www.stellarinfo.com/blog/fix-sql-database-recovery-pending-state-issue/>

Husk:

Højreklik db \ properties, noter db fil navn & db log navn.

Solution 1: Mark database in Emergency mode, detach the main database and re-attach it

ALTER DATABASE [DBName] SET EMERGENCY;

ALTER DATABASE [DBName] set multi\_user

EXEC sp\_detach\_db ‘[DBName]’

EXEC sp\_attach\_single\_file\_db @DBName = ‘[DBName]’, @physname = N'[mdf path]’

These commands will cause the server to get rid of the corrupt log and build a new one automatically.

Solution 2:

Take Offline

Rename til ny kopi

Kopier fra gl lokation

Take online

dbcc Checkdb ,repair

senere fjern kopien

use <dbnavn>;

--dbcc Checkdb(<dbnavn>)

ALTER DATABASE <dbnavn> SET RECOVERY full

ALTER DATABASE <dbnavn> SET EMERGENCY

ALTER DATABASE <dbnavn> SET single\_USER WITH ROLLBACK IMMEDIATE

dbcc Checkdb(<dbnavn>,repair)

ALTER DATABASE <dbnavn> SET MULTI\_USER WITH ROLLBACK IMMEDIATE

## Tag en genstridig Database offline

USE master;

GO

-- Wipeout any user connections, switching user mode on db.

ALTER DATABASE [ CitrixXenApp76Monitoring]

--SET SINGLE\_USER --wrong connection may grab single user

WITH ROLLBACK IMMEDIATE;

GO

ALTER DATABASE [foo] SET SINGLE\_USER WITH ROLLBACK IMMEDIATE

GO

ALTER DATABASE [foo] SET SINGLE\_USER

GO

-- Flip user mode back (all other than my connction is lost/or in error)

ALTER DATABASE [ CitrixXenApp76Monitoring]

SET MULTI\_USER;

GO

-- Take the Database Offline

ALTER DATABASE [ CitrixXenApp76Monitoring] SET OFFLINE WITH ROLLBACK IMMEDIATE

GO

-- Sæt online igen

ALTER DATABASE[XYZ] SET ONLINE

***GO***

-- Rename Database

**ALTER DATABASE** [XYZ] MODIFY **NAME** [XYZnew]

https://www.mssqltips.com/sqlservertip/1891/steps-to-rename-a-sql-server-database/

## --Kopi af Database fra prod til test

Brug p:\sqldba\ps1\sql\GenBkRstScript.sql

Ret fillok, vælg blandt mulige

Ret dbnavn

Ret Append fra \_Anonym til \_Test

Buffer 90000

Output til Text

Kør

Del 1 på ny fane laver backup

Ret Datastore privilegier hvis nødvendige.

Ret del 2 til med datanavn & filnavn

List db users på test server

Kør del 2 på test server

Ret filnavne til

Ret evt. Users tilbage.

Omdøb fil navne

USE [mTIME\_STPS\_TEST]

GO

ALTER DATABASE [mTIME\_STPS\_TEST] MODIFY FILE (NAME=N'mTIME\_STPS\_PROD', NEWNAME=N'mTIME\_STPS\_Test')

GO

USE [mTIME\_STPS\_TEST]

GO

ALTER DATABASE [mTIME\_STPS\_TEST] MODIFY FILE (NAME=N'mTIME\_STPS\_PROD\_log', NEWNAME=N'mTIME\_STPS\_Test\_log')

GO

## The state an SQL database is in can be checked by running the following query:

SELECT name, state, state\_desc from sys.databases

## Moving SQL Server disk to new Hardware / Storage

Add new storage disk to server named x:NewDisk

Check [Databases & Disk fil navn / placering](#_Find_Databaser_&) gem resultat i Excel ark.

Check at Pagefile : (Performance Options: SystemPropertiesPerformance.exe \ Advanced) ikke har extent på disk der skal flyttes, og flyt hvis den har

Stop alle MSSQL services fra MS SQL Configuration Manager (samme version som SQL instansen), sæt SQL Server Services til Service\StartMode:Manual (højreklik & Properties)

Stop alle services:

* Check\_MK
* TSM \*
* nxlog (Logpoint overførselsprogram)
* SQL \* (der ikke er med i sql config manager)
* Volume Shadow Copy

Tøm Recycle Bin

Klargør Robocopy Job, f.eks. Powershell: Kopier folderstruktur inkl. Acl.

Robocopy.exe T:\ Q:\ /E /Copy:S /IS /IT

$Sourcepath1 = "\\srv-mssql-06t.sst.dk\R$\"

$Destinationpath1 = "\\srv-mssql-06t.sst.dk\x$\"

Robocopy $Sourcepath1 $Destinationpath1 /ZB /J /E /PURGE /copyall /sec /xo /r:3 /w:6 /mt:128 /ndl /np /v /log:c:\robocopy\RC\_mssql06t\_xo1.txt

Robocopy $Sourcepath1 $Destinationpath1 /MIR /copyall /Z /XA:H /W:5

{Kør job, check log, ret fejl, ret lognavn } repeat indtil job er kørt fejlfrit igennem.

Brug Disk Management: diskmgmt.msc til at omdøbe gl-disk til gl-disk\_old med temp drevbogstav, og ny disk til korrekt navn & drevbogstav.

Start MSSQL Services fra MS SQL Configuration Manager.

Start MS SQL Server Management Studio, connect til alle instances, og check at der ikke er [(suspect) eller (recovery Pending)](#_Database_(suspect)_eller) databaser (og fix disse)

Sæt Start Mode til Automatic for MSSQL Services via MS SQL Configuration Manager.

Genstart Server, og check at alt er korrekt oppe.

Dokumenter alt undervejs.

Efter 1 uge fjern de gamle diske.

# Oprettelse af Certifikat til Server, til brug af f.eks. HTTPS fra SSRS eller IIS

Vælg administrations server hvor PowerShell er sat korrekt op. (f.eks. s-adm02-p.dksund.dk)

Start powershell som administrator

Net use p: \\dksund.dk\koncern\it-inf\p-drev\install

Cd P:\SqlDba\ps1\cert\

.\New-WebSiteCertificate.ps1 <Servernavn>

Giver

Directory: C:\Users\adm\_mive\Desktop

Mode LastWriteTime Length Name

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

-a---- 26-02-2019 16:18 3659 s-skypedb01-p.dksund.dk.pfx

Certifikat genereret til server: s-skypedb01-p.dksund.dk

Kopier Certifikat fra desktop til <Server> og installer via ii -kommandoen

Certifikat genereret til Netbios name s-skypedb01-p

Gem Certifikat & password i Secret Server: <Certifikat password>

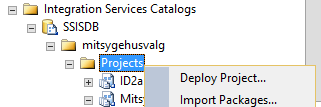
Password skal bruges for at installere certifikatet!

<skjult password men klartekst ved kørsel>

## Deploy SSIS package.

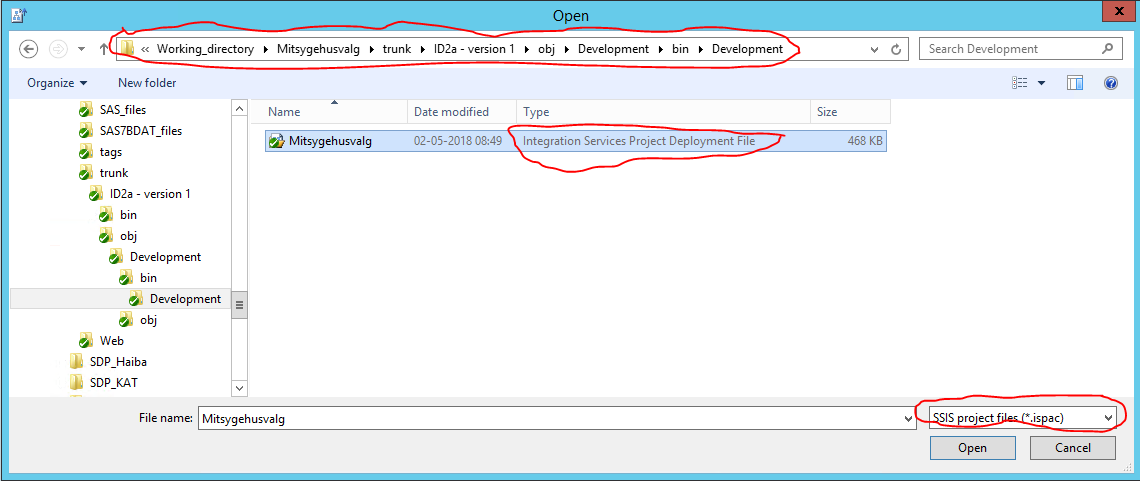
Åbn MS SQL Server Management studio\Integration Services Catalog\SSISDB\<System>\Projects

Højreklik Projects vælg Deploy Project



Vælg Source: Project deployment file

Vælg path til working directory og SSIS Projekt deployment fil. (Check dato!!!)

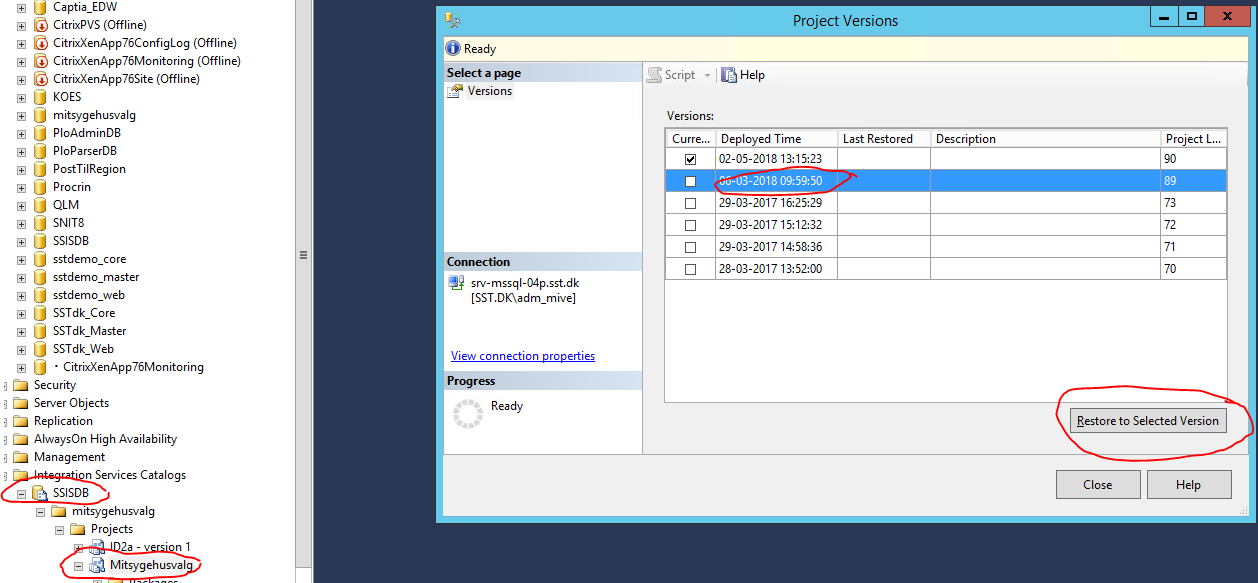


## Rollback SSIS package.

Man laver roll back af en SSIS pakke ved at gendeploye den foregående.

Åbn MS SQL Server Management studio\Integration Services Catalog\SSISDB\<System>\Projects\<Projekt>

Højreklik <Projekt> vælg Versions



Klik på foregående version, tryk på ”Restore to Selected Version”

## Check\_MK SSAS Cube age via Powershell.

# 1 SSAS Cube checket:

Param ($ServerName='localhost:51099')

$loadInfo = [Reflection.Assembly]::LoadWithPartialName("Microsoft.AnalysisServices")

$server = New-Object Microsoft.AnalysisServices.Server

$server.connect($ServerName)

if ($server.name -eq $null) {

Write-Output ("Server ‘{0}’ not found" -f $ServerName)

Write-Output ( "3 HaibaCubeAge 0 HaibaCuben er UNKNOWN timer gammel (max 48 timer) Cant connect server")

break

}

$database = $server.Databases.Item('HAIBA\_OUTPUT\_P01')

$cube = $database.cubes[“HAIBA”]

#Write-Output ( "Cube Processe Last update: {0}" -f $cube.LastProcessed)

$mday = Get-Date

#Write-Output ( "Yesterday: {0}" -f $mday.AddDays(-1))

$minsbetween = (NEW-TIMESPAN –Start $cube.LastProcessed –End $mday).TotalHours

If($cube.LastProcessed -lt $mday.AddDays(-2) ) {

#Write-Output ( " Too Old" )

Write-Output ( "2 HaibaCubeAge float={0} HaibaCuben er {0} timer gammel (max 48 timer)" -f [Math]::round($minsbetween,2))

} Else {

#Write-Output ( " Too Current" )

Write-Output ( "0 HaibaCubeAge float={0} HaibaCuben er {0} timer gammel (max 48 timer)" -f [Math]::round($minsbetween,2))

}

#Stop-Transcript

# Generisk SSAS Cube Check:

Param ($ServerName='localhost:51099')

$loadInfo = [Reflection.Assembly]::LoadWithPartialName("Microsoft.AnalysisServices")

$server = New-Object Microsoft.AnalysisServices.Server

$server.connect($ServerName)

if ($server.name -eq $null) {

Write-Output ("Server ‘{0}’ not found" -f $ServerName)

Write-Output ( "3 HaibaCubeAge 0 HaibaCuben er UNKNOWN timer gammel (max 48 timer) Cant connect server")

break

}

foreach ($database in $server.Databases){

$mday = Get-Date

foreach ($cube in $database.cubes)

{

#Write-Output ( "Yesterday: {0}" -f $mday.AddDays(-1))

$minsbetween = (NEW-TIMESPAN –Start $cube.LastProcessed –End $mday).TotalHours

If($cube.LastProcessed -lt $mday.AddDays(-2) )

{

#Write-Output ( " Too Old" )

Write-Output ( "2 ${database}\_\_\_${cube} float={0} Cuben er {0} timer gammel (max 48 timer)" -f [Math]::round($minsbetween,2))

}

Else

{

#Write-Output ( " Too Current" )

Write-Output ( "0 ${database}\_\_\_${cube} float={0} Cuben er {0} timer gammel (max 48 timer)" -f [Math]::round($minsbetween,2))

}

}

}

#Stop-Transcript

# SSAS Analysis Services Adm

<https://www.mssqltips.com/sqlservertutorial/3600/sql-server-analysis-services-administration-tutorial/>

<https://www.mssqltips.com/sql-server-tip-category/150/analysis-services-administration/>

<https://www.mssqltips.com/sqlservertutorial/3607/sql-server-analysis-services-server-security/>

<https://technet.microsoft.com/en-us/library/ms174893(v=sql.90).aspx>

<https://docs.microsoft.com/en-us/sql/analysis-services/server-properties/security-properties>

<https://docs.microsoft.com/en-us/sql/analysis-services/instances/grant-server-admin-rights-to-an-analysis-services-instance>

## Multidimensional Modeling (Adventure Works Tutorial)

<https://docs.microsoft.com/en-us/sql/analysis-services/multidimensional-modeling-adventure-works-tutorial?view=sql-analysis-services-2017>

## Tabular modeling (1400 compatibility level)

<https://docs.microsoft.com/en-us/sql/analysis-services/tutorial-tabular-1400/as-adventure-works-tutorial?view=sql-analysis-services-2017>

## SSAS Kuber ældre end 2 dage ( CubeAge)

$ServerName=(Get-Item env:computername).value

$ssas=get-service | select status, name,displayname | Where {$\_.Name -like "\*MSSQLServerOLAP\*"}

if($ssas.status -eq 'Running') {

$ssasInstanceName=$ServerName

$f1=Get-NetFirewallRule | where { $\_.Enabled -eq 'True' -and $\_.Direction -eq 'Inbound' -and $\_.DisplayName -Like 'SQL\*Analysis\*'}

If (($f1.DisplayName).length -gt 0) {

# Find firewall port

$f2=$f1 | Get-NetFirewallPortFilter

$ssasInstanceName+=":"

$ssasInstanceName+=$f2.LocalPort

}

$loadInfo = [Reflection.Assembly]::LoadWithPartialName("Microsoft.AnalysisServices")

$server = New-Object Microsoft.AnalysisServices.Server

$server.connect($ssasInstanceName)

$mday = Get-Date

foreach ($database in $server.Databases){

foreach ($cube in $database.cubes){

$HourDiff = (NEW-TIMESPAN –Start $cube.LastProcessed –End $mday).TotalHours

$status=$cube.state

$KubeNavn= "${ServerName}:\_${database}:\_${cube}" -replace(" ","\_")

# status navn graf=val;warn;crit;min;max Human message

# status navn graf=val|graf2=val2|graf3=val3 Human message

Write-Output ( "P $KubeNavn cubeage={0};42;48 Cuben er {0} timer gammel [$Status] (Warn 42, Crit 48 timer)" -f [Math]::round($HourDiff,2))

}

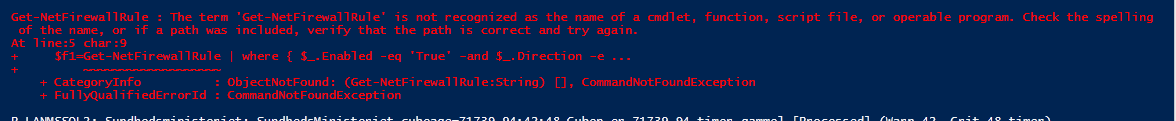
}

} Else {

Write-Host "No SSAS on $ServerName"

}

Fejl kan ske: (lanmssql2)



## SSAS Kube Proccessering fejl

Blog: If the issue is that you get a timeout error after 60 minutes of processing, then a quick fix is to change the ExternalCommandTimeout setting in SSAS to 36000 (or 10 hours).

If SSAS cube processing is causing SQL Server to use a tremendous amount of tempdb space, then that may be an indication that SQL is doing a ton of parallelism. You may try setting the SQL Server "max degree of parallelism" setting on the server to something other than 0... maybe try 4 to begin with and see if that helps.

Another thing I would suggest is to use Glenn Berry's troubleshooting script and specifically look at SQL Server wait stats. If a high percentage of waits are CX\_PACKET then that would be another indication that you need to lower max degree of parallelism as mentioned above:

http://sqlserverperformance.wordpress.com/2010/12/29/updated-sql-2005-and-2008-diagnostic-queries/

Hope some of the suggestions in this thread help.

http://artisconsulting.com/Blogs/GregGalloway

Blog: Hi,

500 million rows is a huge data and you should definitely consider partitioning your cube. Once the partitions is been done, you can process required partition of the cube instead of whle cube at a time. As partitions will be having less number of rows it will be quicker and will require less Temp DB space.

Here are some good articals on how to partition you cube,

http://www.bidn.com/blogs/briankmcdonald/bidn-blog/1212/creating-analysis-services-partitions-using-bids

http://www.sqlservercentral.com/articles/Analysis+Services+(SSAS)/70282/

Cheers, Ashish

Explosive Temdb growth can be caused by aggregations(Group by), complex joins in views, complex queries, order by clauses i queries(and so on). It is probably about how you feed data into SSAS or any other non SSAS activities when you process the cube.

What you can do is to to put Tempdb on a large and fast dedicated hard-drive. Before you start the SSAS processing you can shrink the tempdb. Each time you restart the server the Tempdb is dropped and recreated with a smaller space.

Thomas Ivarsson

# SSRS Reporting Services Adm

<https://docs.microsoft.com/en-us/sql/reporting-services/report-server-sharepoint/configuration-and-administration-of-a-report-server>

https://technet.microsoft.com/en-us/library/bb522825(v=sql.105).aspx

<https://www.mssqltips.com/sql-server-tip-category/104/reporting-services-administration/>

<https://docs.microsoft.com/en-us/sql/reporting-services/report-server/configure-and-administer-a-report-server-ssrs-native-mode>

<https://docs.microsoft.com/en-us/sql/reporting-services/security/grant-user-access-to-a-report-server>

<https://docs.microsoft.com/en-us/sql/reporting-services/install-windows/find-reporting-services-product-key-ssrs?view=sql-server-2017>

<https://docs.microsoft.com/en-us/sql/reporting-services/install-windows/install-reporting-services?view=sql-server-2017>

<https://blog.sqlauthority.com/2013/07/24/sql-server-installing-sql-server-data-tools-and-ssrs/>

<https://www.mssqltips.com/sqlservertip/5237/installing-sql-server-reporting-services-2017/>

<https://www.mssqltips.com/sqlservertip/4852/sql-server-reporting-services-standalone-installation/>

<https://www.mssqltips.com/sqlservertip/4020/sql-server-reporting-services-general-best-practices/>

<https://blog.devoworx.net/2017/02/04/install-and-configure-ssrs/>

<https://www.advancedinstaller.com/user-guide/ssrs-deployment.html>

## Reporting Services Configuration Manager

RSC Connection: angiv server & instans: Connect

RSC Status: info om Server, instance, edition, version db-name, mode, status & mulighed for start /stop

RSC Service Account: vælg Builtin (ReportServer) eller ad: gMSA service account til at starte ReportServeren…

RSC WebService URL: Vælg virtual Directory, IP, port, SSL certificate, SSL port og resulterende URL

RSC Database: Vælg SQL Server, DB, RS mode, DB Credential, Login, password

RSC ReportManager/Web Portal URL: vælg Virtual Directory, URLs

RSC E-mail settings: vælg Sender Address, Delivery Method (SMTP), SMTP Server

RSC Execution Account: Vælg Account & Password

RSC Encryption Key: Backup, Restore, Change, Delete (Demands Reconfiguration afterwards)

RSC Scale-out Deployment (Distribueret Reporting Services)

# Microsoft R Services

R-services handler jo om data-analyse og machine learning og her taler vi om Microsofts implementering af R (som er et gammelt sprog)

R-services giver (med vores viden pt.) flere fordele:

Det afvikles på SQL Serveren, hvorfor data ikke skal flyttes over netværket.

Der kan afvikles R-kode i SSMS.

Memory and disk will be managed by your instance of SQL Server (in-memory database) etc, etc…..

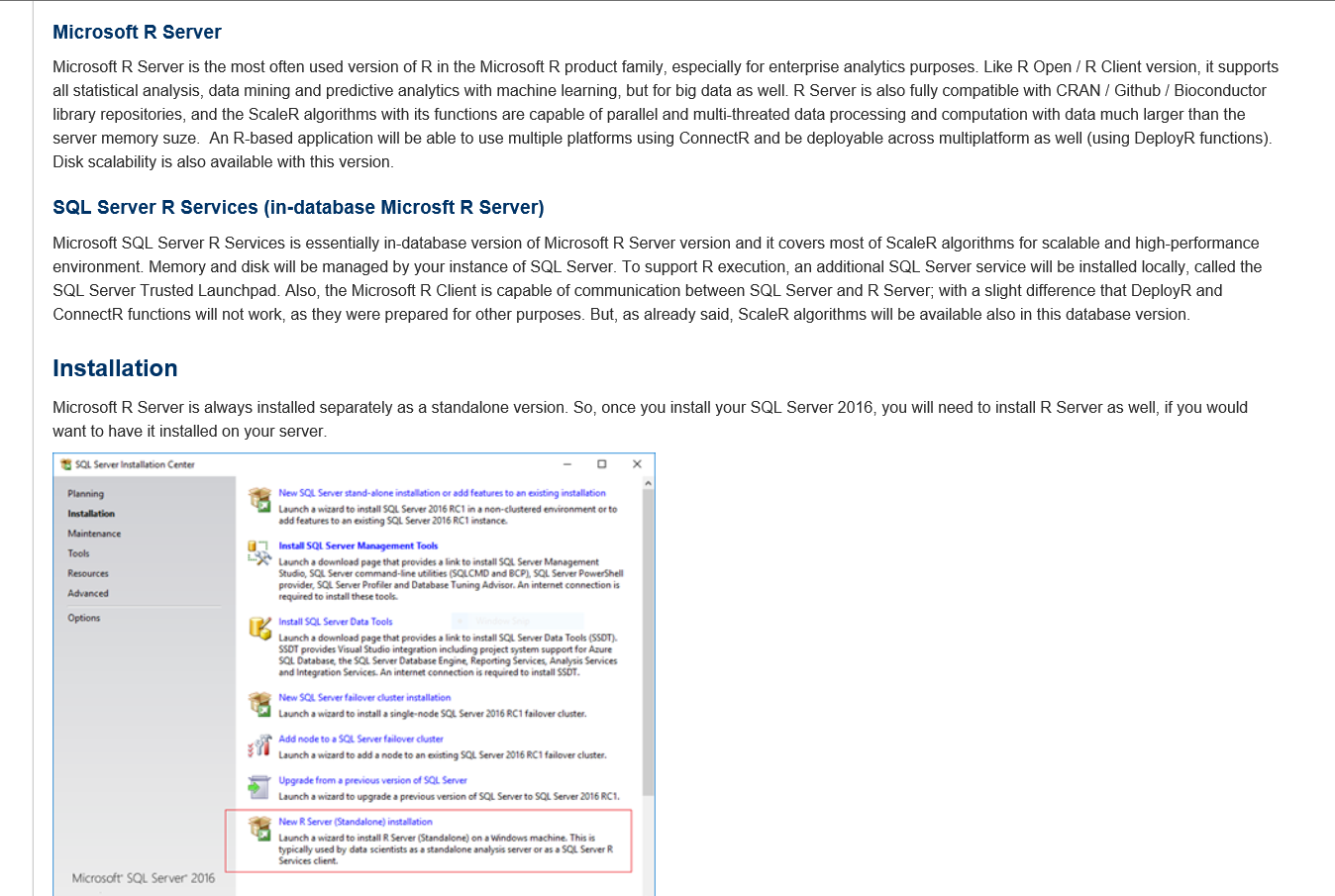
“With R Services available in SQL Server 2016, Microsoft is making the R language available for more flexible data analysis, making sharing data insights much easier, and overcoming memory (RAM) limitations. This mean the integration of R into SQL Server makes analysis on larger datasets, real time OLTP analysis and any kind of big data analysis (document system such as Hadoop, Teradata and others) much easier than ever before.”

“In SQL Server 2016, Microsoft introduced SQL Server R Services, a feature that supports enterprise-scale data science by integrating the R language with SQL Server database engine.”

Introduction to Microsoft R Services in SQL Server 2016 (non-Microsoft side dog)

<http://www.sqlservercentral.com/articles/Microsoft/145393/>

“Using R Services and R Server brings better data insights with fast data computation and little (if any at all) data movement. Especially with R Server, performance will be enhanced and the size of your dataset will no longer be limited by memory. This tool is perfect for any kind of organization and corporate environment, that wish to shorten the time for creating daily and custom statistical analysis in in later parts we will also see in which areas R as a statistical and predictive tool can be used.”



“Introducing Machine Learning with SQL Server” (den kaldes Machine Learning Server I SQL Server 2017)

<https://docs.microsoft.com/en-us/machine-learning-server/r/concept-what-is-sql-server-r-services>

# DMV

### Sys.dm\_io\_virtual\_file\_stats

Use select \* from to inquire about the SQL Server workload.

### sys.dm\_tran\_locks

### sys.dm\_os\_waiting\_tasks.

This DMV displays the list of tasks that are currently waiting together with the wait type, the wait duration, the resource that the task is waiting for, and other details.

### sys.dm\_os\_wait\_stats.

This DMV displays the total wait time for each wait type.

### sys.dm\_exec\_session\_wait\_stats.

This DMV displays the same data as the previous DMV but filtered for a single active user session.

## Check Backup jobs

Viser hvilke backup jobs der er aktive med start-tid, %Færdig og estimeret sluttid

use [master]

SELECT session\_id as SPID, command, a.text AS Query, start\_time, percent\_complete, dateadd(second,estimated\_completion\_time/1000, getdate()) as estimated\_completion\_time

FROM sys.dm\_exec\_requests r CROSS APPLY sys.dm\_exec\_sql\_text(r.sql\_handle) a

WHERE r.command in ('BACKUP DATABASE','RESTORE DATABASE', 'BACKUP LOG')

### sys.dm\_exec\_requests

### sys.dm\_exec\_sql\_text

### sys.dm\_server\_services

Denne DMV viser info om Service accounts

### Master.sys.sql\_modules

Hent opsætnings parametre

SELECT uses\_ansi\_nulls, uses\_quoted\_identifier, \*

FROM sys.sql\_modules

--WHERE object\_id = object\_id('SampleProcedure')

## Monitor tempdb:

### sys.dm\_db\_file\_space\_usage

dynamic management view to monitor the disk space that the files are using. Additionally, to monitor the page allocation or deallocation activity in tempdb at the session or task level, you can use the

### sys.dm\_db\_session\_space\_usage & sys.dm\_db\_task\_space\_usage

# Faste Parametre

Select suser\_name() as WhoAmI --Viser udførende burger navn (& domain)

Select DEFAULT\_DOMAIN() --Viser AD Domain (uden .dk mm)

Select @@Servername --Viser sql server navn (no domain)

Select @@Servicename --Viser sql server Instans navn

Select @@version --Viser sql server Instans version (incl. servicepack)

Select @@LANGUAGE --Viser sql server sprog

Select @@SPID --Viser sql server Instans version (incl. servicepack)

Select @@CURSOR\_ROWS, @@FETCH\_STATUS, @@ROWCOUNT, @@TOTAL\_READ, @@TOTAL\_WRITE, @@TRANCOUNT

--Viser param ved brug af cursor & fetch

Select CONVERT (varchar, SERVERPROPERTY('collation')) AS 'Server Collation'

Select DATABASEPROPERTYEX('dba\_db','Collation') AS 'DB Collation'

Select name DBName,collation\_name,recovery\_model from sys.databases AS 'DB Collation'

Select Serverproperty('Productversion') -- viser sql version i tal: 13.0.4502.0

‘ProductMajorVersion’: -- ver#: 13

‘ProductMinorVersion’: -- ver#: 13

'Productlevel': -- Servicepack (or blank): SP1

'ProductUpdateLevel': -- Cumulative Update (or blank): CU9

'Edition': -- Edition: Enterprise Edition: Core-based Licensing (64-bit)

'InstanceName': -- blank for mssqlserver default instance

'MachineName': -- Hostname

'ServerName': -- Windows server and instance

'InstanceDefaultDataPath': -- path to data: r:\...data

'InstanceDefaultLogPath' -- path to Log: s:\...data

‘Collation’ -- SQL Server Arbejdssprog.

‘CollationID’, ‘ComputerNamePhysicalNetBIOS’, ‘IsIntegratedSecurityOnly’, ‘IsSingleUser’, ‘ProcessID’

SELECT DEFAULT\_DOMAIN(),SERVERPROPERTY('MachineName'),(SELECT distinct InstanceName FROM DBA\_DB.dbo.InstanceInfo)

Giver f.eks.: DKSUND S-MSSQL01-T MSSQLSERVER

SELECT Convert(varchar,SERVERPROPERTY('MachineName'))+'.'+DEFAULT\_DOMAIN()+'\'+(SELECT distinct InstanceName FROM DBA\_DB.dbo.InstanceInfo)

Giver tilsvarende: S-MSSQL01-T.DKSUND\MSSQLSERVER

invoke-sqlcmd -ServerInstance s-inf-kms-01p -query "Select Serverproperty('Edition')”

invoke-sqlcmd -ServerInstance s-inf-kms-01p -query "Select Serverproperty('ProductLevel')”

invoke-sqlcmd -ServerInstance s-inf-kms-01p -query "Select Serverproperty('Productversion')”

invoke-sqlcmd -ServerInstance s-inf-kms-01p -query "Select Serverproperty('ProductLevel')”

invoke-sqlcmd -ServerInstance s-inf-kms-01p -query "Select Serverproperty('ProductLevel')”

invoke-sqlcmd -ServerInstance s-inf-kms-01p -query "Select Serverproperty('ProductLevel')”

invoke-sqlcmd -ServerInstance s-inf-kms-01p -query "Select Serverproperty('ProductLevel')”

invoke-sqlcmd -ServerInstance s-inf-kms-01p -query "Select Serverproperty('ProductLevel')”

invoke-sqlcmd -ServerInstance s-inf-kms-01p -query "Select Serverproperty('ProductLevel')”

Serverproperty('ProductUpdateLevel')"

## Extended properties

USE [master]

GO

EXEC [DBA\_DB].sys.sp\_addextendedproperty @name=N'Db\_FingerPrint', @value=N'0x16896513621030'

GO

select value from DBA\_DB.sys.extended\_properties where NAME = 'Db\_FingerPrint'

USE [master]

GO

EXEC [DBA\_DB].sys.sp\_addextendedproperty @name=N'Db\_BS', @value=N'MSSQL' --Business Service

GO

EXEC [DBA\_DB].sys.sp\_addextendedproperty @name=N'Db\_Ejer', @value=N'Mive' --managed By

GO

EXEC [DBA\_DB].sys.sp\_addextendedproperty @name=N'Db\_gdpr', @value=N'Nej' --Persondata

GO

EXEC [DBA\_DB].sys.sp\_addextendedproperty @name=N'Db\_ SuppGrp', @value=N'Database-Bi Assignment' --Support Group

GO

EXEC [DBA\_DB].sys.sp\_addextendedproperty @name=N'Db\_Descr', @value=N'MSSQL Collecting System & db information Depository' --Description

GO

if Not exists(select 1 from sys.extended\_properties where NAME = 'Db\_Ejer')

Begin

EXEC [DBA\_DB].sys.sp\_addextendedproperty @name=N'Db\_Ejer', @value=N'Mive'

End

if Not exists(select 1 from sys.extended\_properties where NAME = 'Db\_BS')

Begin

EXEC [DBA\_DB].sys.sp\_addextendedproperty @name=N'Db\_BS', @value=N'MSSQL'

End

if Not exists(select 1 from sys.extended\_properties where NAME = 'Db\_gdpr')

Begin

EXEC [DBA\_DB].sys.sp\_addextendedproperty @name=N'Db\_gdpr', @value=N'Nej'

End

if Not exists(select 1 from sys.extended\_properties where NAME = 'Db\_SuppG')

Begin

EXEC [DBA\_DB].sys.sp\_addextendedproperty @name=N'Db\_SuppG', @value=N'Database-Bi Assignment'

End

if Not exists(select 1 from sys.extended\_properties where NAME = 'Db\_Descr')

Begin

EXEC [DBA\_DB].sys.sp\_addextendedproperty @name=N'Db\_Descr', @value=N'MSSQL Collecting System & db information Depository'

End

select \* from sys.extended\_properties

-- Db\_FingerPrint

if exists(select 1 from sys.extended\_properties where NAME = 'Db\_Ejer')

Begin

select \* from sys.extended\_properties

End

else Begin

Select Value='N/A'

End

exec sp\_addextendedproperty

@name = N'SNO'

,@value = N'Testing entry for Extended Property'

,@level0type = N'Schema', @level0name = 'dbo'

,@level1type = N'Table', @level1name = 'mytest'

,@level2type = N'Column', @level2name = 'sno'

go

[Working with SQL Server Extended Properties (mssqltips.com)](https://www.mssqltips.com/sqlservertip/5384/working-with-sql-server-extended-properties/)

[Scripting the Description of Database Tables Using Extended Properties - Simple Talk (red-gate.com)](https://www.red-gate.com/simple-talk/devops/database-devops/scripting-description-database-tables-using-extended-properties/)

[Query Extended Properties from Multiple SQL Server Databases (mssqltips.com)](https://www.mssqltips.com/sqlservertip/5982/query-extended-properties-from-multiple-sql-server-databases/)