## Restoring a Database

USE [master]

RESTORE DATABASE [AdventureWorks2014] FROM DISK = N'C:\DBA\AdventureWorks2014.bak'

WITH FILE = 1, MOVE N'AdventureWorks2014\_Data' TO

N'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\AdventureWorks2014\_Data.mdf',

MOVE N'AdventureWorks2014\_Log' TO

N'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\AdventureWorks2014\_Log.ldf'

## Backing up Database

BACKUP DATABASE [AdventureWorks2014] TO DISK = N'C:\DBA\AdventureWorksBackup2' WITH NOFORMAT, NOINIT, NAME = N'AdventureWorks2014-Full Database Backup', SKIP, NOREWIND, NOUNLOAD, STATS = 10

GO

## Change Recovery Model

ALTER DATABASE [AdventureWorks2014] SET RECOVERY FULL WITH NO\_WAIT

GO

## Backing up in SSMS

BACKUP DATABASE [AdventureWorks2014]

TO DISK = N'C:\DBA\AdventureWorksBackup2' WITH DIFFERENTIAL

, NOFORMAT, NOINIT, NAME = N'AdventureWorks2014-Full Database Backup'

, SKIP, NOREWIND, NOUNLOAD, STATS = 10

BACKUP LOG [AdventureWorks2014] TO DISK = N'C:\DBA\AdventureWorksBackup2'

WITH NOFORMAT, NOINIT, NAME = N'AdventureWorks2014-Full Database Backup',

SKIP, NOREWIND, NOUNLOAD, STATS = 10

## Using NORECOVERY and RECOVERY

restore database [AdventureWorks2014] with recovery

BACKUP LOG [AdventureWorks2014] TO DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\Backup\AdventureWorks2014\_LogBackup\_2018-07-31\_18-35-51.bak' WITH NOFORMAT, NOINIT, NAME = N'AdventureWorks2014\_LogBackup\_2018-07-31\_18-35-51', NOSKIP, NOREWIND, NOUNLOAD, STATS = 5

RESTORE DATABASE [AdventureWorks2014backup2] FROM DISK = N'C:\DBA\AdventureWorksBackup2' WITH FILE = 2, MOVE N'AdventureWorks2014\_Data' TO N'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\AdventureWorks2014backup2\_Data.mdf', MOVE N'AdventureWorks2014\_Log' TO N'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\AdventureWorks2014backup2\_Log.ldf', NORECOVERY, NOUNLOAD, STATS = 5

RESTORE DATABASE [AdventureWorks2014backup2] FROM DISK = N'C:\DBA\AdventureWorksBackup2' WITH FILE = 5, NORECOVERY, NOUNLOAD, STATS = 5

RESTORE LOG [AdventureWorks2014backup2] FROM DISK = N'C:\DBA\AdventureWorksBackup2' WITH FILE = 6, NORECOVERY, NOUNLOAD, STATS = 5

RESTORE LOG [AdventureWorks2014backup2] FROM DISK =

N'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\Backup\AdventureWorks2014\_LogBackup\_2018-07-30\_22-21-00.bak'

WITH FILE = 1, NORECOVERY, NOUNLOAD, STATS = 5

RESTORE LOG [AdventureWorks2014backup2] FROM DISK =

N'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\Backup\AdventureWorks2014\_LogBackup\_2018-07-30\_22-28-32.bak'

WITH FILE = 1, NOUNLOAD, STATS = 5

## 13d. Recover from a corrupted Drive

ALTER DATABASE [AdventureWorks2014backup3] SET SINGLE\_USER WITH ROLLBACK IMMEDIATE

DBCC CHECKDB ([AdventureWorks2014backup3], REPAIR\_ALLOW\_DATA\_LOSS)

ALTER DATABASE [AdventureWorks2014backup3] SET MULTI\_USER;

# Logins and Users

## 17c. Create Login accounts

CREATE LOGIN [DESKTOP-KBB0EE5\SQLTest] FROM WINDOWS WITH DEFAULT\_DATABASE=[AdventureWorks2014]

GO

## 17b. Secure the SQL Server with server roles

ALTER SERVER ROLE [sysadmin] ADD MEMBER [DESKTOP-KBB0EE5\SQLTest]

GO

## 17e. Create and Maintain User-Defined Server Roles

CREATE SERVER ROLE [MyServerRole2]

GO

ALTER SERVER ROLE [MyServerRole2] ADD MEMBER [DESKTOP-KBB0EE5\SQLTest]

GO

use [master]

GO

DENY ALTER ANY LOGIN TO [MyServerRole2]

GO

## 19c. Create Database User Accounts

USE [AdventureWorks2014]

GO

CREATE USER [SQLTest] FOR LOGIN [DESKTOP-KBB0EE5\SQLTest] WITH DEFAULT\_SCHEMA=[dbo]

GO

## Add Users to Fixed Database Roles

ALTER ROLE [db\_datareader] ADD MEMBER [SQLTest]

GO

## Create User Database-Level Roles

USE [AdventureWorks2014]

GO

CREATE ROLE [MyNewDatabaseRole]

GO

USE [AdventureWorks2014]

GO

ALTER ROLE [MyNewDatabaseRole] ADD MEMBER [SQLTest]

GO

use [AdventureWorks2014]

GO

DENY SELECT ON [HumanResources].[Employee] TO [MyNewDatabaseRole]

GO

use [AdventureWorks2014]

GO

GRANT SELECT ON [HumanResources].[Department] TO [MyNewDatabaseRole]

GO

## Schema

USE [AdventureWorks2014]

GO

CREATE SCHEMA [MyNewSchema] AUTHORIZATION [SQLTest]

GO

## 15d. Implement indexes

USE [AdventureWorks2014]

GO

CREATE CLUSTERED INDEX [IX\_NewTable\_ID] ON [dbo].[NewTable]

[ID])

GO

CREATE NONCLUSTERED INDEX [IX\_NewTable\_Color] ON [dbo].[NewTable]

(ColorName] ASC)

INCLUDE ([ObjectName])

GO

DROP INDEX [IX\_NewTable\_Color] ON [dbo].[NewTable]

GO

## 15e. Defrag/Rebuild indexes

USE [AdventureWorks2014]

GO

ALTER INDEX [IX\_NewTable\_ID] ON [dbo].[NewTable] REBUILD WITH (ONLINE = ON)

GO

ALTER INDEX [IX\_NewTable\_ID] ON [dbo].[NewTable] REORGANIZE

GO

select \*

from sys.dm\_db\_index\_physical\_stats(DB\_ID('AdventureWorks2014')

,object\_id(N'[Person].[Address]'),null,null,null) as stats

join sys.indexes as i

on stats.object\_id = i.object\_id and stats.index\_id = i.index\_id

USE [AdventureWorks2014]

GO

ALTER INDEX [PK\_Address\_AddressID] ON [Person].[Address] REORGANIZE WITH ( LOB\_COMPACTION = ON )

GO

ALTER INDEX [AK\_Address\_rowguid] ON [Person].[Address] REORGANIZE WITH ( LOB\_COMPACTION = ON )

GO

ALTER INDEX [IX\_Address\_AddressLine1\_AddressLine2\_City\_StateProvinceID\_PostalCode] ON [Person].[Address] REORGANIZE WITH ( LOB\_COMPACTION = ON )

GO

ALTER INDEX [IX\_Address\_StateProvinceID] ON [Person].[Address] REORGANIZE WITH ( LOB\_COMPACTION = ON )

GO

## 15c. Identify unused indexes

select object\_name(u.object\_id) as [Object Name], \*

from sys.dm\_db\_index\_usage\_stats as U

join sys.indexes as I

on u.object\_id = i.object\_id and u.index\_id = i.index\_id

## 15h. Create Statistics

create statistics St\_NewTable on [dbo].[NewTable] ([ColorName])

where [ColorName] is not null

## 16a. Transfer data

EXEC master.dbo.sp\_detach\_db @dbname = N'AdventureWorks2014backup'

GO

CREATE DATABASE [AdventureWorks2014backup] ON

( FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\AdventureWorks2014backup\_Data.mdf' ),

( FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\AdventureWorks2014backup\_Log.ldf' )

FOR ATTACH

GO

## 16b. Bulk Copy

bcp AdventureWorks2014.dbo.NewTable in “c:\dba\BCPOut.dat” -T -c

bcp AdventureWorks2014.dbo.NewTable2 out “c:\dba\BCPOut.dat” -T -c

## 16c. Bulk Insert

bulk insert [dbo].[FlatFile3]

from 'c:\dba\FlatFile.txt'

with

(FIELDTERMINATOR = ',',

ROWTERMINATOR = '\n',

FIRSTROW = 2

)

## 5b. Administer jobs

USE msdb

GO

select \* from sysjobsteps

select \* from syssessions

select \* from sysjobactivity

select \* from sysjobhistory

select \* from sysschedules

## 5b. Administer Alerts

exec sp\_addmessage 50001, 16, 'I am raising an alert'

select \* from sys.messages

RAISERROR (50001, 16, 1) WITH LOG

# Manage and configure databases

## 6a. Design multiple file groups

CREATE DATABASE [DBAdatabase]

CONTAINMENT = NONE

ON PRIMARY

( NAME = N'DBAdatabase', FILENAME =

N'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\DBAdatabase.mdf'

, SIZE = 102400KB , FILEGROWTH = 65536KB ),

FILEGROUP [Secondary]

( NAME = N'DBAdatabase2', FILENAME =

N'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\DBAdatabase2.ndf'

, SIZE = 102400KB , FILEGROWTH = 65536KB )

LOG ON

( NAME = N'DBAdatabase\_log', FILENAME =

N'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\DBAdatabase\_log.ldf'

, SIZE = 8192KB , FILEGROWTH = 65536KB )

GO

IF NOT EXISTS (SELECT name FROM sys.filegroups WHERE is\_default=1 AND name = N'Secondary')

ALTER DATABASE [DBAdatabase] MODIFY FILEGROUP [Secondary] DEFAULT

GO

## 6c. Moving objects from one filegroup to another

--Creating new Clustered Index on a different filegroup

CREATE CLUSTERED INDEX [ClusteredIndex-20180805-100419] ON [dbo].[tblTable]

(

[Heading1] ASC,

[Heading2] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON)

ON [Third]

GO

--Creating new table and specifying filegroup

create table dbo.tblTable2

(Heading1 int,

Heading2 int)

ON [Third]

--Moving an existing index to a different filegroup

CREATE CLUSTERED INDEX [ClusteredIndex-20180805-100419] ON [dbo].[tblTable]

(

[Heading1] ASC,

[Heading2] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF

, DROP\_EXISTING = ON, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON)

ON [Secondary]

GO

## 13e, 14d-e. Managing and restoring a multi-TB database

BACKUP DATABASE [DBAdatabase] FILEGROUP = N'Secondary' TO

DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\Backup\DBAdatabase.bak'

WITH NOFORMAT, NOINIT, NAME = N'DBAdatabase-Full Database Backup', SKIP, NOREWIND, NOUNLOAD, STATS = 10

GO

RESTORE DATABASE [DBAdatabase] FILE = N'DBAdatabase2'

FROM DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\Backup\DBAdatabase.bak'

WITH FILE = 1, NOUNLOAD, STATS = 10

GO

RESTORE DATABASE [AdventureWorks2014] PAGE='23:34' FROM DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\Backup\AdventureWorks2014\_LogBackup\_2018-07-31\_18-35-51.bak' WITH FILE = 6, NORECOVERY, NOUNLOAD, STATS = 5

## 6h. Manage Log File Growth

select \*

from sys.dm\_db\_log\_space\_usage

select \*

from sys.databases

dbcc shrinkfile(DBAdatabase\_log, 3)

## 6i. DBCC

dbcc shrinkdatabase([DBAdatabase],20)

select \* from sys.database\_files

dbcc shrinkfile(3, truncateonly)

dbcc shrinkfile(DBAdatabase3, emptyfile)

## 19d. Contained login

EXEC sys.sp\_configure N'contained database authentication', N'1'

GO

RECONFIGURE WITH OVERRIDE

GO

## 6d, 19d. Implement and configure contained databases

sp\_configure 'show advanced options', 1

go

RECONFIGURE

GO

EXEC sys.sp\_configure N'contained database authentication', N'1'

GO

RECONFIGURE WITH OVERRIDE

GO

sp\_configure 'show advanced options', 0

go

RECONFIGURE

GO

ALTER DATABASE [DBAdatabase] SET CONTAINMENT = PARTIAL

GO

USE [DBAdatabase]

GO

CREATE USER [ContainedUser] WITH PASSWORD=N'Passw0rd!'

GO

## 6e. Data Compression

USE [AdventureWorks2014]

ALTER INDEX [IX\_Employee\_OrganizationNode] ON [HumanResources].[Employee]

REBUILD PARTITION = ALL WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF,

SORT\_IN\_TEMPDB = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON,

DATA\_COMPRESSION = PAGE)

select \*

from sys.indexes

where object\_name(object\_id)='Department'

exec sp\_estimate\_data\_compression\_savings

'HumanResources', 'Department', 2, NULL, 'PAGE'

## 15k. Column store indexes

CREATE NONCLUSTERED COLUMNSTORE INDEX [NonClusteredColumnStoreIndex-20180805-151345]

ON [dbo].[tblPartition]

(

[DateOfEntry]

)WITH (DROP\_EXISTING = OFF, COMPRESSION\_DELAY = 0)

GO

## 10d. Diagnose blocking, live locking and deadlocking

--Session 1

begin tran

update [dbo].[FlatFile]

set Heading1 = heading1+1

--Session 2

begin tran

update [dbo].[FlatFile2]

set Heading1 = heading1+1

select \*

from [dbo].[FlatFile]

--Session 1

begin tran

update [dbo].[FlatFile2]

set Heading1 = heading1+1

## 10c. Monitor via DMV or other MS product; 10e. diagnose waits

exec sp\_who2

--kill 59

select \*

from sys.databases

select resource\_type, request\_status, request\_mode, resource\_description, request\_session\_id

from sys.dm\_tran\_locks

where resource\_database\_id = 5 and request\_session\_id in (58, 59)

select \*

from sys.dm\_os\_waiting\_tasks

where session\_id in (58, 59)

select \*

from sys.dm\_exec\_requests

## 10f, 10g. Performance detection with built in DMVs

select \* --CPU

from sys.dm\_os\_schedulers

where scheduler\_id <256

select count(database\_id)\*8/1024.0 as [Cache in Mb], database\_id --Buffer Pool/Data Cache

from sys.dm\_os\_buffer\_descriptors

group by database\_id

select \*

from sys.sysperfinfo

where object\_name like 'SQLServer:Buffer Manager%'

order by counter\_name

## 11b. Collect performance data by using System Monitor

diskperf -Y

## 12e. Monitor elevated privileges as well as unsolicited attempts to connect

select prin.name, perm.\*

from sys.server\_permissions as perm

join sys.server\_principals as prin

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

EXEC sys.sp\_configure N'show advanced options', N'1' RECONFIGURE WITH OVERRIDE

GO

EXEC sys.sp\_configure N'common criteria compliance enabled', N'1'

GO

RECONFIGURE WITH OVERRIDE

GO

EXEC sys.sp\_configure N'show advanced options', N'0' RECONFIGURE WITH OVERRIDE

GO

exec xp\_readerrorlog 2

## 4b. Full Text Index

select \*

from [Person].[Address]

where CONTAINS(AddressLine1, 'Drive NEAR Glaze')

## 4c. SSIS Security

USE msdb

go

select \*

from sysssispackages

select \*

from sysssislog

## 4d. Filestream

EXEC sp\_configure filestream\_access\_level, 2

RECONFIGURE

CREATE DATABASE [Filestream]

CONTAINMENT = NONE

ON PRIMARY

( NAME = N'Filestream', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\Filestream.mdf' , SIZE = 8192KB , FILEGROWTH = 65536KB ),

FILEGROUP [FilestreamData] CONTAINS FILESTREAM

( NAME = N'FilestreamData', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\FilestreamData' )

LOG ON

( NAME = N'Filestream\_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\Filestream\_log.ldf' , SIZE = 8192KB , FILEGROWTH = 65536KB )

GO