stellar

BEST PRACTICES GUIDE FOR
SQL BACKUP & RECOVERY PLAN

STELLAR WHITE PAPER | JANUARY 2018

TABLE OF CONTENT

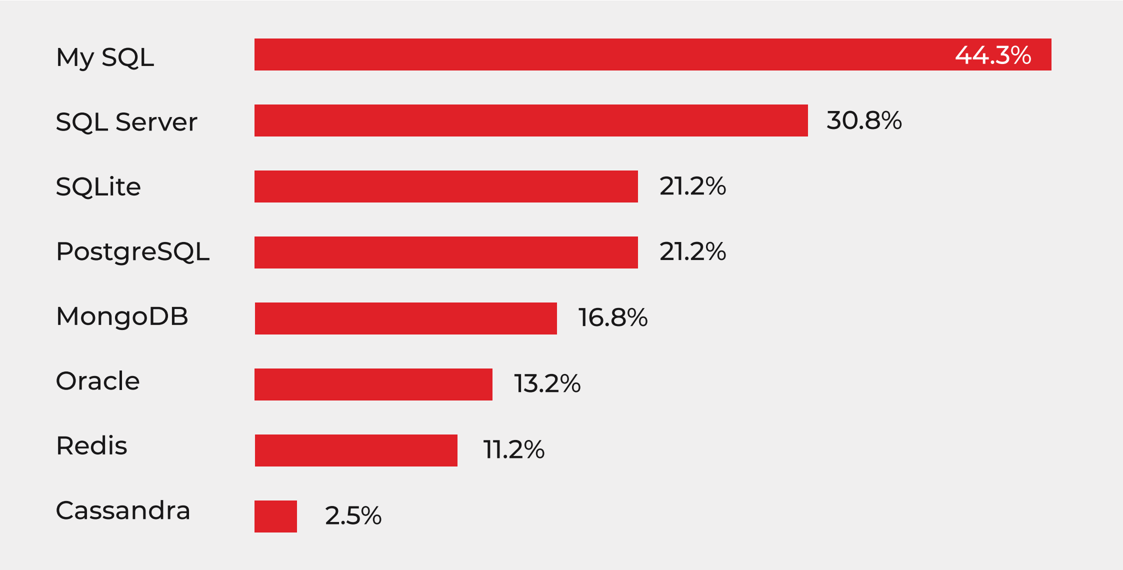
INTRODUCTION

ABOUT MICROSOFT SQL SERVER

Microsoft SQL Server was initially developed by Sybase which later was acquired by Microsoft prior to the 1990s. This was the first introduction of Microsoft into the enterprise-level database management system which was mainly dominated by Oracle, IBM and Sybase. The first version called SQL Server 1.0 for OS/2 platform came around 1989. A later release SQL Server v6.0 in 1992 was available for Windows NT. In 1996, an update was release called v6.5 and then in 1998 the famous SQL Server 7.0 was released.

Since then, Microsoft SQL Server has undergone a lot of development lifecycle with features like Policy-Based Management, Change Data Capture, Transparent Data Encryption, Backup and Data Compression, Resource Governor, AlwaysOn, SQL Clustering, and more.

The chart below shows the growth of the different RDMS systems as of the end of 2017. Microsoft SQL Server has shown tremendous increase in growth.



AN INTRODUCTION OF SQL DATABASE CORRUPTION

Are you ready when this happens? Unfortunately, SQL database corruption is totally inevitable due to either user error or multiple external factors like virus infection, operating system malfunction, and file system corruption. This can render the SQL database to be inaccessible, which not only puts the data at risk but also impedes the business processes and even profits.

This step-by-step guidance will be a great pocket book which discusses common corruption situations and how to repair/retrieve the corrupted data as much as possible. As part of a standard disclaimer with regard to database corruption, please read below as stated by Microsoft.

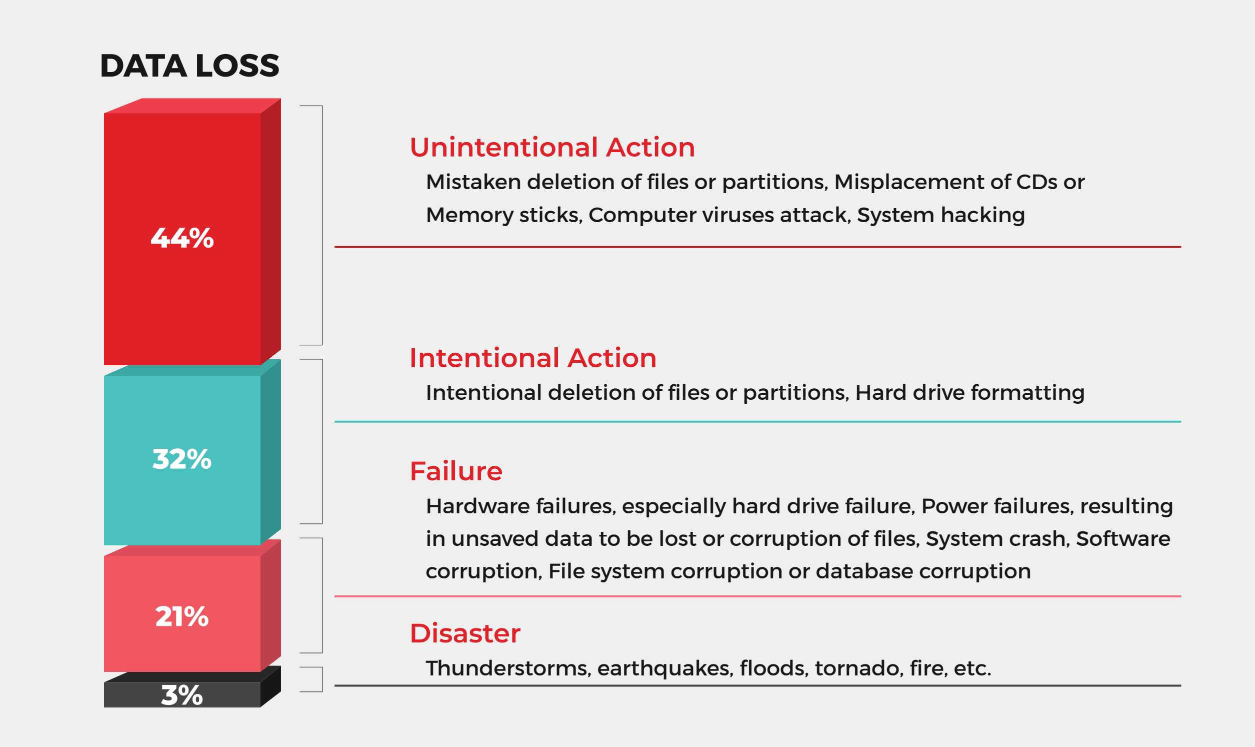
Although Microsoft has written the SQL Server product, Microsoft does not guarantee that databases will never become corrupt or in case of corruption, 100% data can be recovered.

I do however believe that Microsoft has taken maximum caution with the coding of SQL Server to protect your data as much as possible.

REASONS OF SQL DATABASE CORRUPTION

Did you know that 95% causes of database corruption end up being an external issue, which is the underlying layer of SQL Server. The list below ranks the most common root causes of a SQL database corruption.

- A hardware failure like Disk Subsystem, Controller, CPU, or Memory Module(s). SQL Server and all DBMS are highly dependent on the I/O for reading and writing data. Any fault in the disk subsystem, communication channels or disk drivers can get the databases to be corrupted.
- Third-party applications drivers or firmware bugs can cause the database to be corrupted as well.
- It is also common that kernel drivers and software like antivirus cause database corruption issues. Some of these software can take kernel handle of database files which are in use by the SQL instance. If an antivirus is to be installed on a database server; please follow the necessary requirements to prevent this issue.



Unintentional Action
Mistaken deletion of files or partitions, Misplacement of CDs or Memory sticks, Computer viruses attack, System hacking

Intentional Action
Intentional deletion of files or partitions, Hard drive formatting

Failure
Hardware failures, especially hard drive failure, Power failures, resulting in unsaved data to be lost or corruption of files, System crash, Software corruption, File system corruption or database corruption

Disaster
Thunderstorms, earthquakes, floods, tornado, fire, etc.

SOLUTIONS TO REPAIR

DATABASE RESTORE FROM THE BACKUP

The SQL Server backup and restore feature ensures that your critical data stored in the SQL Server databases is not lost, should any catastrophic events were to happen to the database. For you to minimize the potential risk of losing data, there is that constant need to back up your databases on a regular basis per the business needs. A well-scheduled backup and restore strategy not only prevents data loss but allows a smooth operation in restoring back the database. Always remember, a backup and restore strategy is unworthy if not tried and tested by restoring the set of backups to verify that they would work in times of disaster recovery.

TAKING BACKUP OF THE DATABASE

Are there any other true solutions in protecting your data without the incorporation of high availability solutions? The only solution is by backing up your databases, running dry test restores on your backups, and sending copies of the backups to a safe and secure disaster recovery location should there be any moment of potential catastrophic data loss. Good news right? You will never fall into the dilemma of what solution would be the best to protect your data.

True valid backups of a database can help you recover data from many types of failures, such as:

- User errors, for example, deleting records from a table or dropping the entire table.
- Hardware failures like a damaged disk drive or server crashing.
- Natural disasters like flooding, fire, or earthquakes.

NOTE: Backups also come in handy for routine database administrative tasks like setting up HADR solutions like Log Shipping, database mirroring, AlwaysOn AG, also moving a database from one server to another, and archiving.

METHODS TO TAKE BACKUP

Backing up a SQL Server database can be performed mainly in two different ways which is by using the SQL Server native tools or a third-party software. This paper is not going to cater for the recommended option or comparison between the two. The entire demonstrations will be using the SQL Native tools as these are widely used and can be easily accessible. Examples of backing up by using SQL Native tools includes TSQL, SQL Server Management Studio, SQL Operations Studio, PowerShell, and even Command Prompt. All these options are free applications that would not require additional purchasing or licensing.

CREATING BACKUP

In creating or restoring SQL Server database backups, this paper would illustrate two methods of the SQL Native tools. This will be using TSQL and SQL Server Management Studio (SSMS). These two methods are the most common native ways that DBAs perform backups of the database.

FULL DATABASE

A full database backup backs up the whole database which includes part of the transaction log so that the full database can be recovered during a restore. Full database backups represent the database at the time the backup was completed. Full backups can be performed on all types of recovery models which is SIMPLE, FULL or BULK-LOGGED. Remember, a restore of a full backup is identical to the state of the original database when the backup was finished minus any uncommitted transactions.

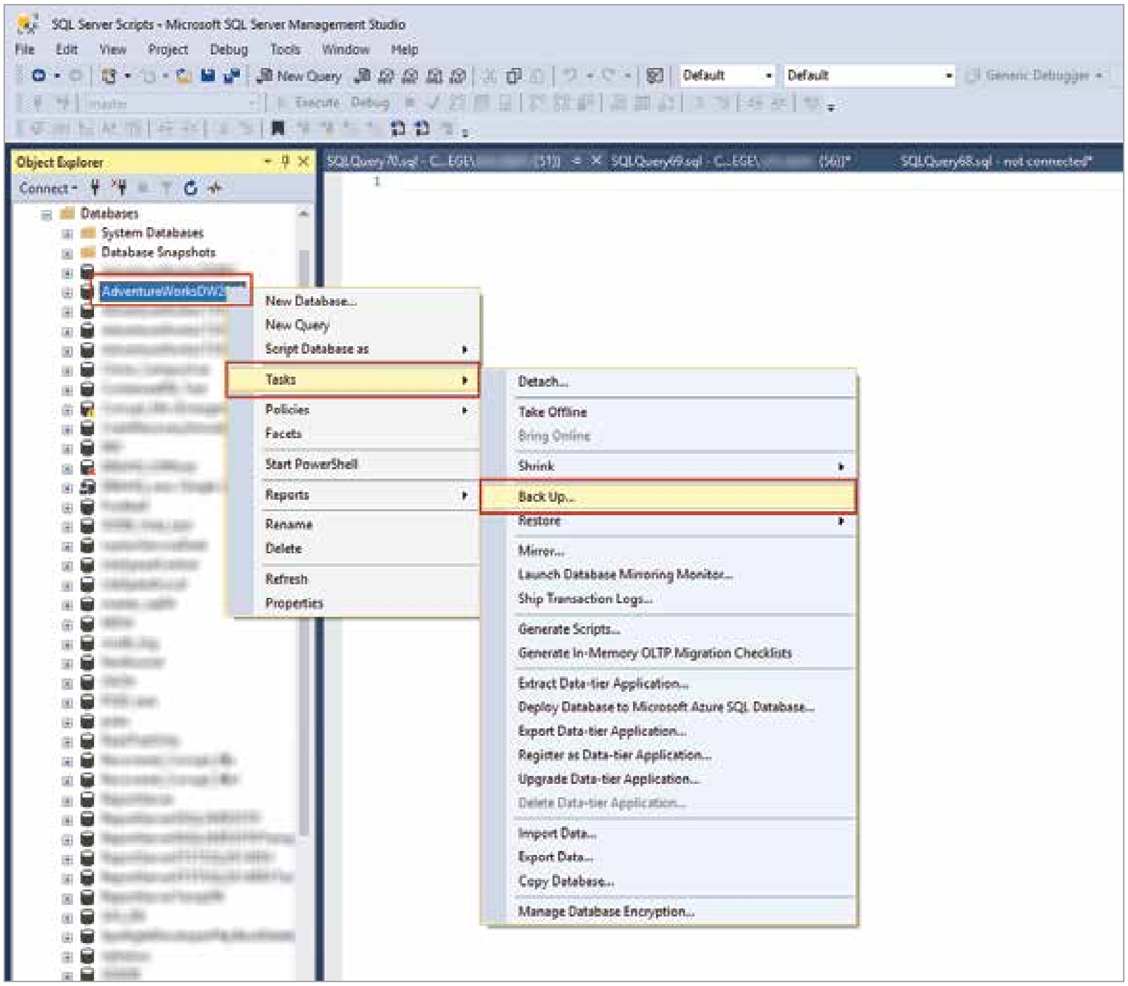
USING TSQL

Using Sample Database AdventureWorks2012 for backup demonstration.

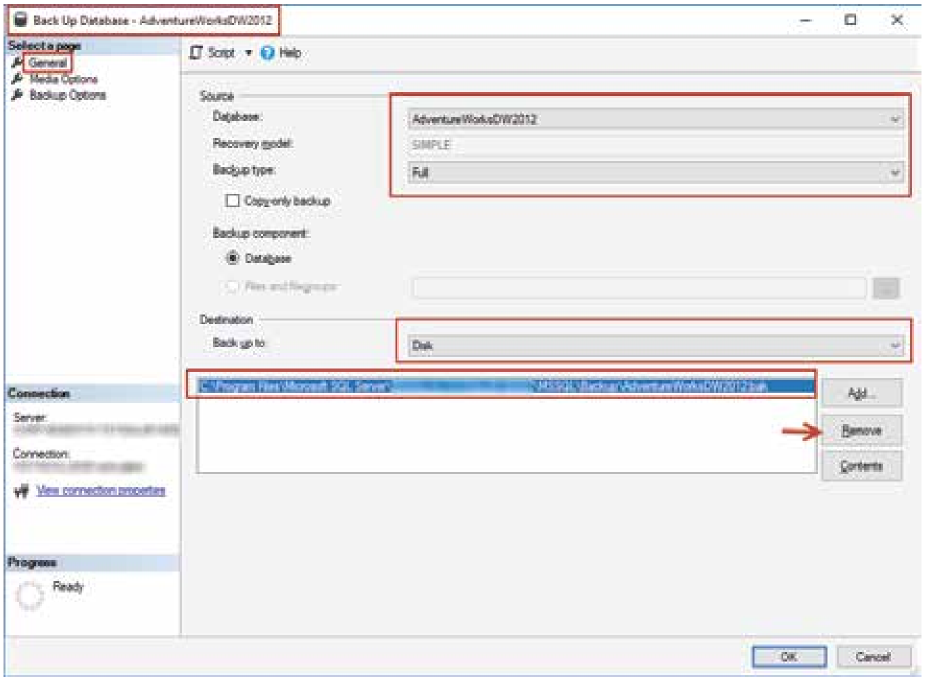
```
USE master;
GO
BACKUP DATABASE AdventureWorks2012
TO DISK = 'G:\MSSQLBackups\AdventureWorks2012_062818.Bak'
WITH FORMAT, MEDIANAME = 'G_BackupShare', NAME = 'Full Backup of AdventureWorks2012';
GO
```

USING SSMS

- Connect to the required SQL server instance by using the appropriate authentication. Using Object Explorer Pane, click the server name to expand the server tree.
- Expand Databases and click the user database to be backed up, or if backing up system databases then expand System Databases and select the appropriate system database. See screenshot for details.
- Right-click the desired database, select Tasks, and then click Back Up. The backup database dialog box appears.


SQL Server Scripts - Microsoft SQL Server Management Studio
File Edit View Project Debug Tools Window Help
New Query
Default
Default
Generic Debugger
Execute
Object Explorer
SQLQuery70.sql - C...EGE\ (51))
SQLQuery69.sql - C...EGE\ (56))*
SQLQuery68.sql - not connected*
Connect
Databases
System Databases
Database Snapshots
AdventureWorksDW2
New Database...
New Query
Script Database as
Tasks
Policies
Facets
Start PowerShell
Reports
Rename
Delete
Refresh
Properties
Detach...
Take Offline
Bring Online
Shrink
Back Up...
Restore
Mirror...
Launch Database Mirroring Monitor...
Ship Transaction Logs...
Generate Scripts...
Generate In-Memory OLTP Migration Checklists
Extract Data-tier Application...
Deploy Database to Microsoft Azure SQL Database...
Export Data-tier Application...
Register as Data-tier Application...
Upgrade Data-tier Application...
Delete Data-tier Application...
Import Data...
Export Data...
Copy Database...
Manage Database Encryption...

- For the General page, there is the Database drop-down list to verify the database name. Optionally, you can select a different database from the list.
- The Recovery model text box is for reference only. You can perform a database backup for any recovery model (FULL, BULK_LOGGED, or SIMPLE). In the Backup type drop-down list, select Full.
- For Backup component, select the Database radio button. If there is already a destination path as shown below. You have the option to either add additional paths or remove and add a new desired path.
- In this scenario, we removed the pre-existing path.



Back Up Database - AdventureWorksDW2012

Source

Database: AdventureWorksDW2012

Recovery model: SIMPLE

Backup type: Full

□ Copy-only backup

Backup component:

● Database

○ Files and filegroups

Destination

Back up to: Disk

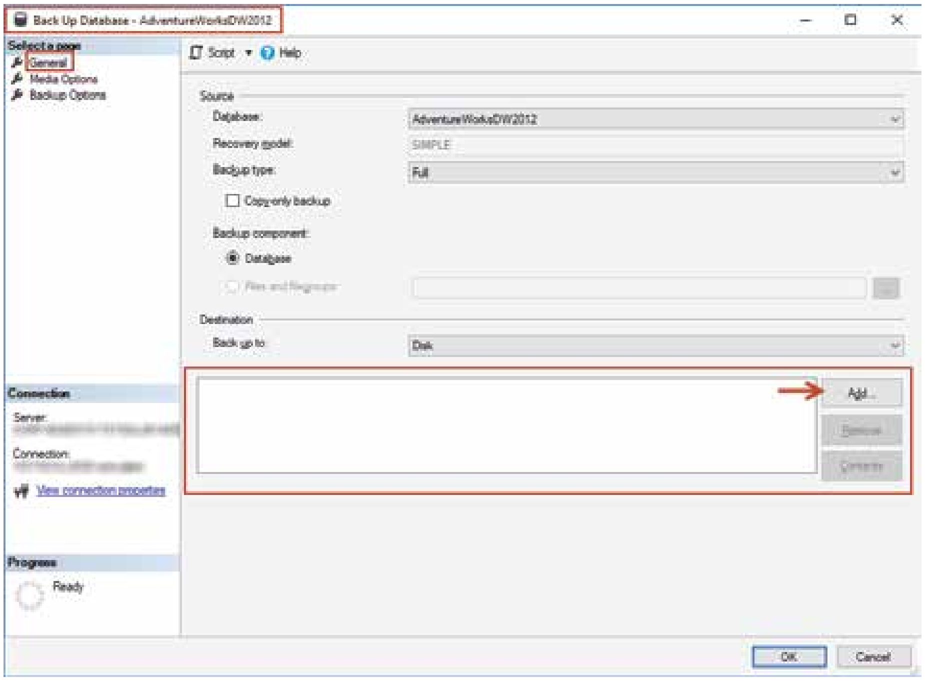
C:\Program Files\Microsoft SQL Server\[redacted]\MSSQL\Backup\AdventureWorksDW2012.bak

Server: [redacted]

Connection: [redacted]

Progress: Ready

- Click on ‘ADD’ to select a desired destination path.



Back Up Database - AdventureWorksDW2012

Source

Database: AdventureWorksDW2012

Recovery model: SIMPLE

Backup type: Full

□ Copy-only backup

Backup component:

● Database

○ Files and filegroups

Destination

Back up to: Disk

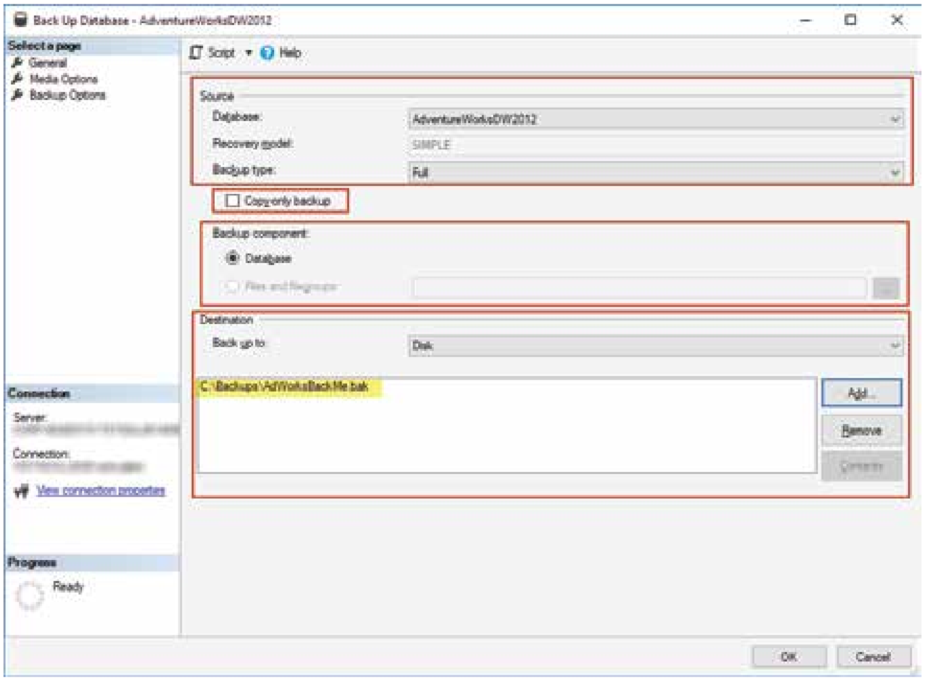
Server: [redacted]

Connection: [redacted]

Progress: Ready

- As an option, you can check the Copy-only backup checkbox to create a copy-only backup instead. A copy-only backup is a SQL Server backup that is independent of the sequence of conventional set of SQL Server backups.

NOTE: For more information, see Copy-Only Backups (SQL Server). A copy-only backup is not available for the Differential backup type.



Back Up Database - AdventureWorksDW2012

Select a page
- General
- Media Options
- Backup Options

Source
Database: AdventureWorksDW2012
Recovery model: SIMPLE
Backup type: Full

☐ Copy-only backup

Backup component:
● Database
○ Files and filegroups:

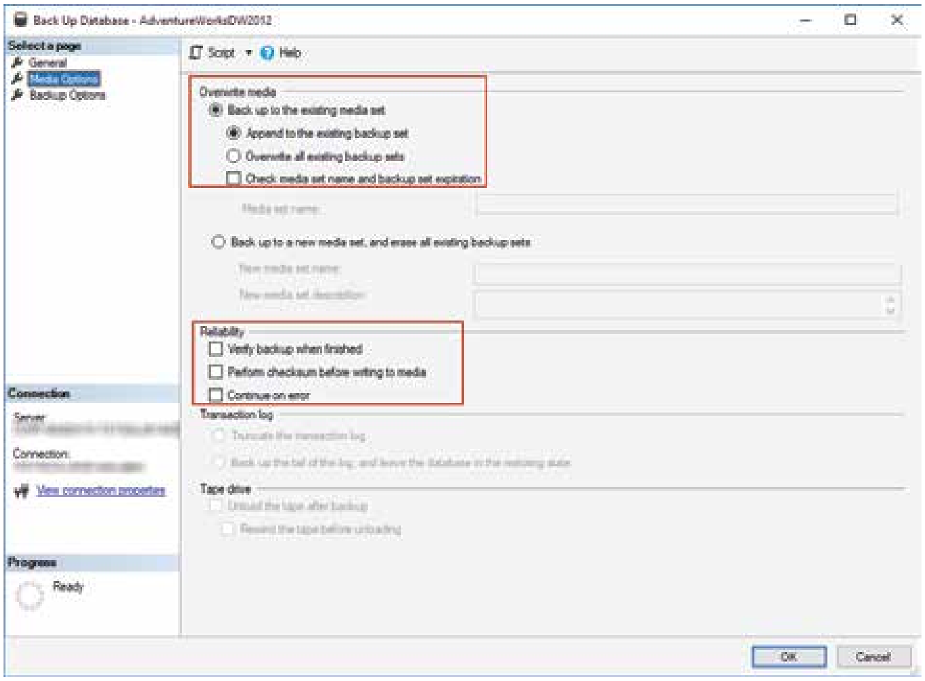
Destination
Back up to: Disk

C:\Backups\AdWorksBackMe.bak

Connection
Server:
[redacted]
Connection:
[redacted]

Progress
Ready

- You can view the Media options by selecting it on the 'Select a Page' Pane.
- There are options on this page to either Overwrite media or Append to existing media.
- The Reliability section performs checks like verify the backup when finished, perform checksum before writing to media, and finally continue even when an error is encountered.



Back Up Database - AdventureWorksDW2012

Select a page
- General
- Media Options
- Backup Options

Overwrite media
● Back up to the existing media set
● Append to the existing backup set
○ Overwrite all existing backup sets
☐ Check media set name and backup set expiration
Media set name:
○ Back up to a new media set, and erase all existing backup sets
New media set name:
New media set description:

Reliability
☐ Verify backup when finished
☐ Perform checksum before writing to media
☐ Continue on error

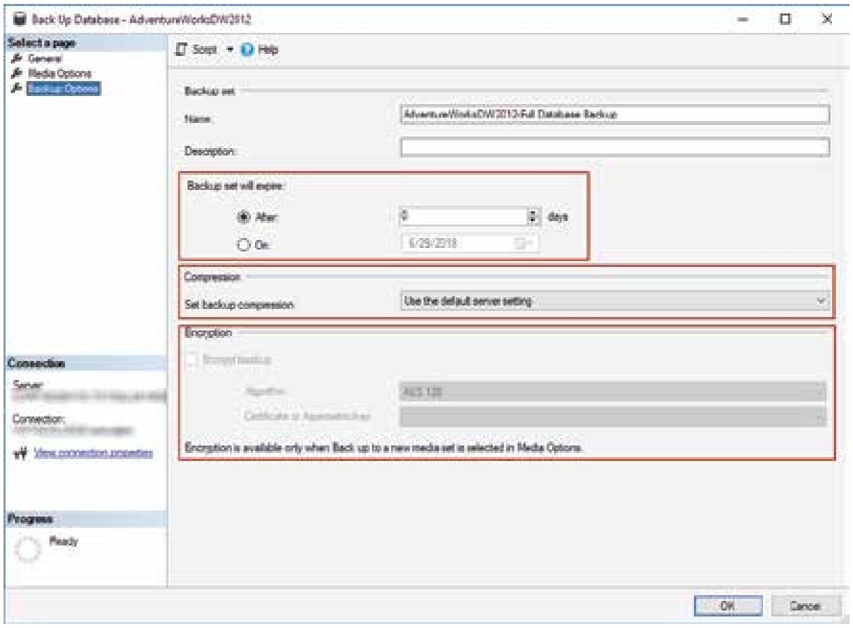
Transaction log
○ Truncate the transaction log
○ Back up the tail of the log, and leave the database in the restoring state

Tape drive
☐ Unload the tape after backup
☐ Rewind the tape before unloading

Connection
Server:
[redacted]
Connection:
[redacted]

Progress
Ready

- The last Page 'Backup Options' provides the functionality to name the Backup Set and a description.
- You also can define the expiration date for the backup set.
- Compression and encryption settings can also be made on this page.



Backup set

Name: AdventureWorksDW2012-Full Database Backup

Description:

Backup set will expire:

● After: 0 days

○ On: 6/29/2018

Compression

Set backup compression: Use the default server setting

Encryption

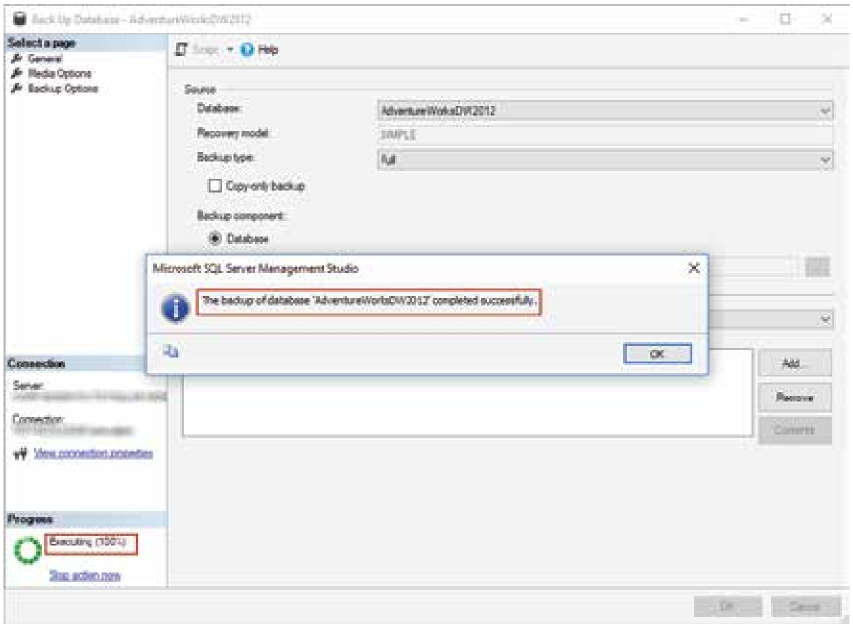
□ Encrypt backup

Algorithm: AES 128

Certificate or Asymmetric key:

Encryption is available only when Back up to a new media set is selected in Media Options.

- Once all appropriate configurations have been completed. Click on OK to start the backup execution.
- Upon successful completion of the backup, SSMS would notify a success dialog box and the Progress status would be at a 100%



Source

Database: AdventureWorksDW2012

Recovery model: SIMPLE

Backup type: Full

□ Copy-only backup

Backup component:

● Database

The backup of database 'AdventureWorksDW2012' completed successfully.

Executing (100%)

DIFFERENTIAL BACKUP

- A differential backup is based on the most recent and previous full data backup. A differential backup captures only the data that has changed since that full backup. The full backup upon which a differential backup is based is known as the base of the differential. Full backups, except for copy-only backups, can serve as the base for a series of differential backups, including database backups, partial backups, and file backups. The base backup for a file differential backup can be contained within a full backup, a file backup, or a partial backup.

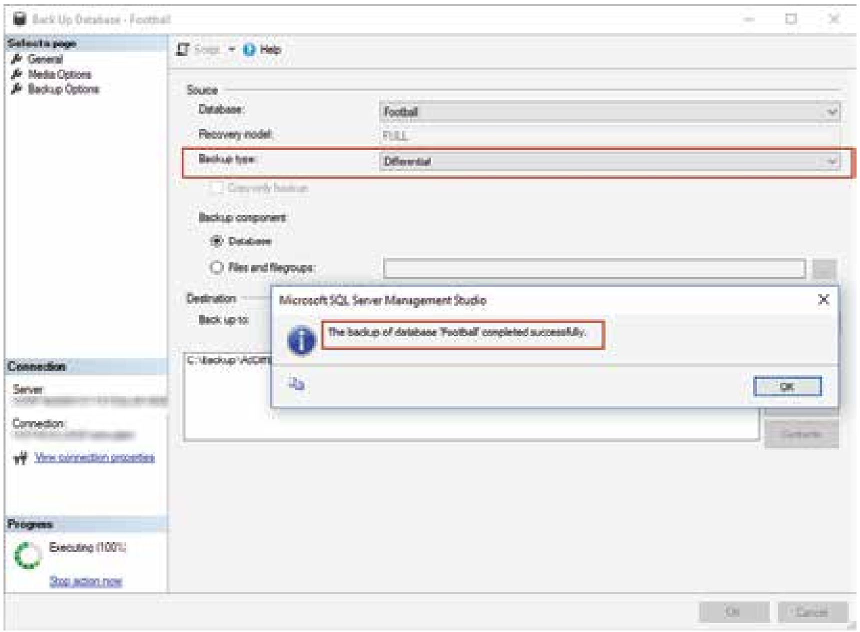
USING TSQL

- Using Sample Database Football for backup demonstration.

```
USE master;
GO
BACKUP DATABASE [Football] TO DISK = N'C:\Backup\AdDiffBackMe.bak' WITH
DIFFERENTIAL, NOFORMAT, NOINIT,  NAME  = N'Football-Full Database Backup' ,
SKIP, NOREWIND, NOUNLOAD, STATS = 10
GO
```

USING SSMS

- Apply same steps as shown above for FULL backup. The main difference here is the Backup type should be changed to Differential.
- Once all appropriate configurations have been completed. Click OK to start the backup execution.
- Upon successful completion of the backup, SSMS would notify a success dialog box and the Progress status would be at 100%



Back Up Database - Football

Source

Database: Football

Recovery model: FULL

Backup type: Differential

□ Copy-only backup

Backup component

● Database

○ Files and filegroups:

Destination

Back up to:

C:\Backup\AdDiff

Microsoft SQL Server Management Studio

The backup of database 'Football' completed successfully.

Progress

Executing (100%)

FILE AND FILE GROUPS

SQL Server allows the database files or filegroups to be backed up due to either the database size or performance requirements.

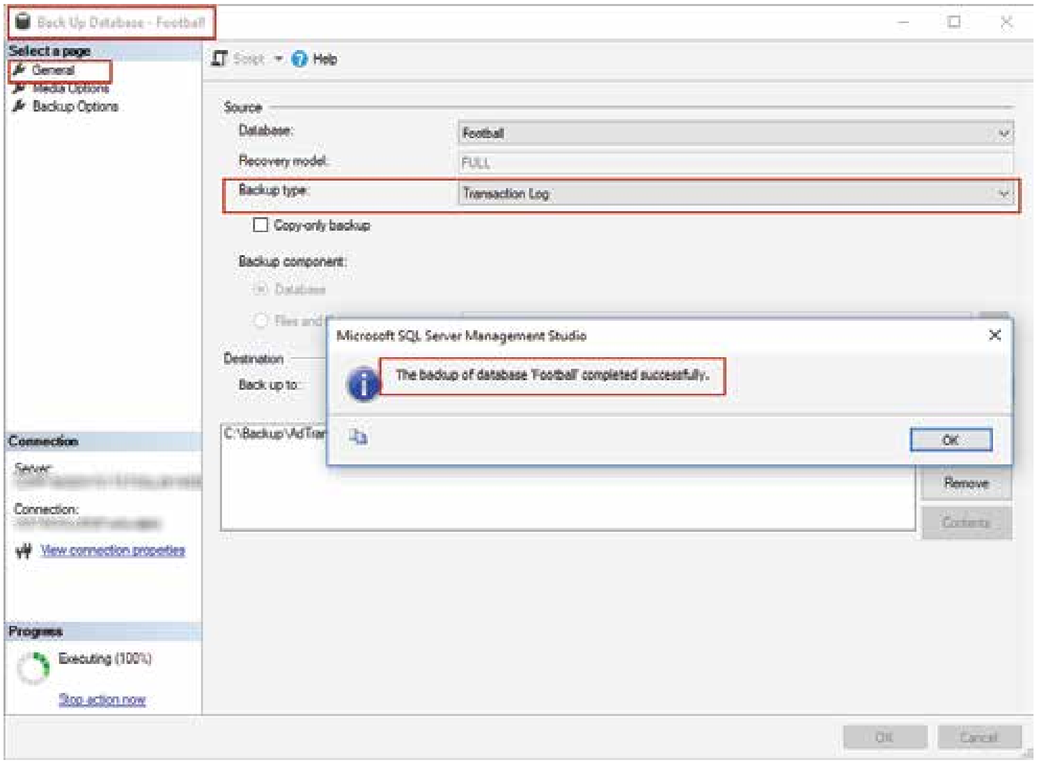
USING TSQL

Using Sample Database Football for file and file groups backup demonstration.

```
USE Football;
GO
BACKUP DATABASE [Football] FILEGROUP = N'POST_SECONDARY'
TO  DISK = N'C:\Backup\AdFileGroup.bak' WITH NOFORMAT, NOINIT,
NAME = N'Football-Full Database Backup' , SKIP, NOREWIND, NOUNLOAD, STATS = 10
GO
```

USING SSMS

- Apply same steps as shown above for FULL backup. The main difference here is the Backup type should be changed to Transaction Log.
- Once all appropriate configurations have been completed. Click OK to start the backup execution.
- Upon successful completion of the backup, SSMS would notify a success dialog box and the Progress status would be at 100%



Source

Database: Football

Recovery model: FULL

Backup type: Transaction Log

☐ Copy-only backup

Backup component:

● Database

○ Files and

Destination

Back up to:

C:\Backup\AdTran

Server:

Connection:

Executing (100%)

Microsoft SQL Server Management Studio

The backup of database 'Football' completed successfully.

OK

FILE AND FILE GROUPS

SQL Server allows the database files or filegroups to be backed up due to either the database size or performance requirements.

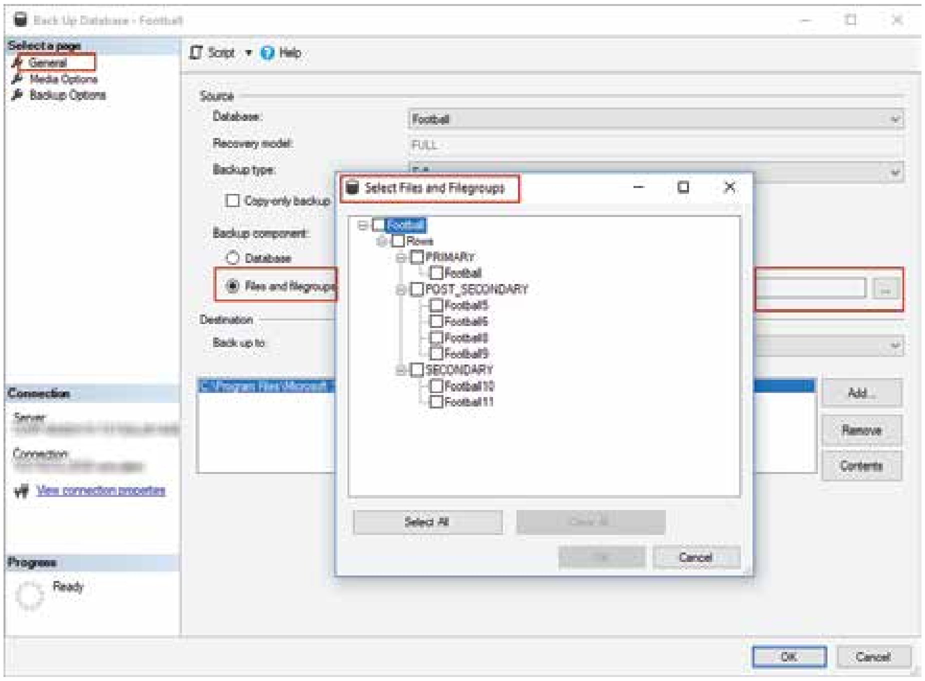
USING TSQL

Using Sample Database Football for file and file groups backup demonstration.

```
USE Football;
GO
BACKUP DATABASE [Football] FILEGROUP = N'POST_SECONDARY'
TO  DISK = N'C:\Backup\AdFileGroup.bak' WITH NOFORMAT, NOINIT,
NAME = N'Football-Full Database Backup' , SKIP, NOREWIND, NOUNLOAD, STATS = 10
GO
```

USING SSMS

- Apply same steps as shown above for FULL backup. The main difference here selecting the radio button Files and filegroups under the Backup component.
- Backup type could either be FULL or DIFFERENTIAL for file or filegroups backup.



Database: Football

Recovery model: FULL

☐ Copy-only backup

Backup component:

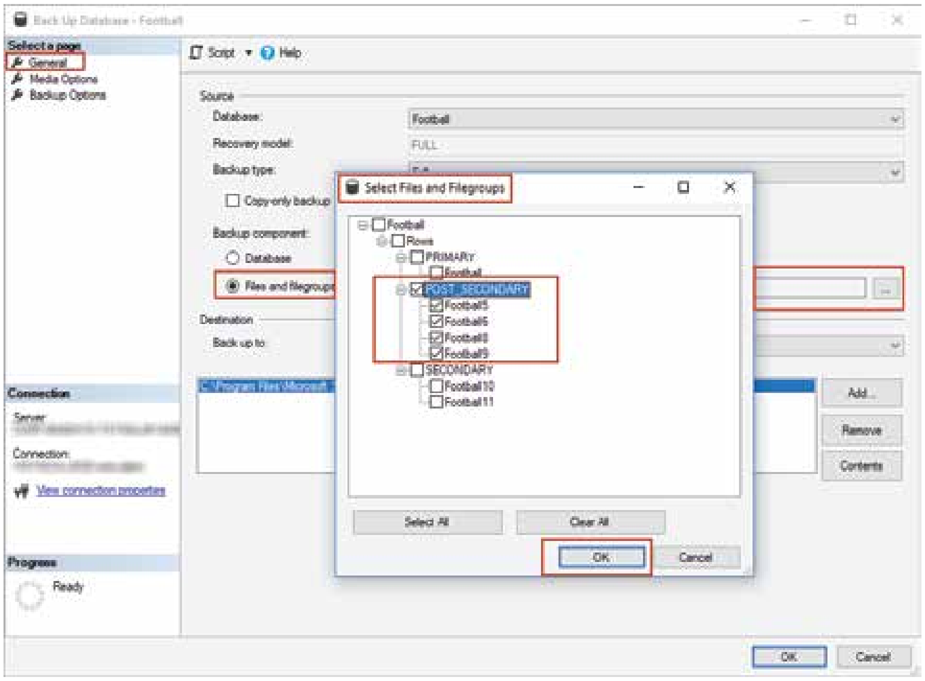
○ Database

● Files and filegroups

Back up to: C:\Program Files\Microsoft

☐ Football
☐ Rows
☐ PRIMARY
☐ Football
☐ POST_SECONDARY
☐ Football5
☐ Football6
☐ Football8
☐ Football9
☐ SECONDARY
☐ Football10
☐ Football11

- In the Select Files and Filegroups dialog box, select the files and filegroups you want to back up. You can select one or more individual files or check the box for a filegroup to automatically select all the files in that filegroup.
- Click OK once completed with selections.



Database: Football

Recovery model: FULL

☐ Copy-only backup

Backup component:

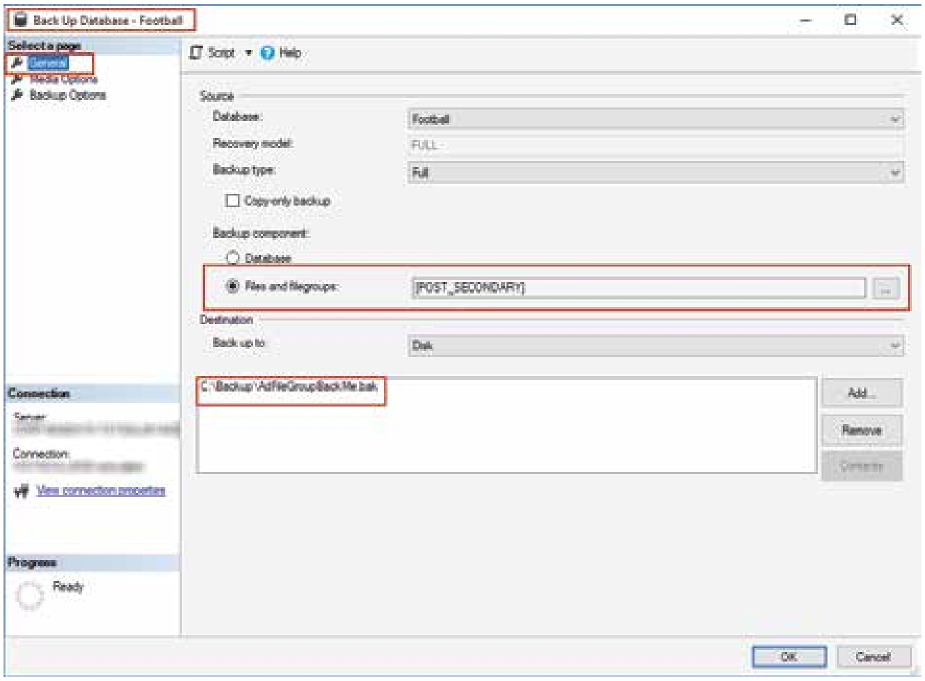
○ Database

● Files and filegroups

Back up to: C:\Program Files\Microsoft

☐ Football
☐ Rows
☐ PRIMARY
☐ Football
☑ POST_SECONDARY
☑ Football5
☑ Football6
☑ Football8
☑ Football9
☐ SECONDARY
☐ Football10
☐ Football11

- Specify the destination of the backup as always.



Back Up Database - Football

Select a page: General, Media Options, Backup Options

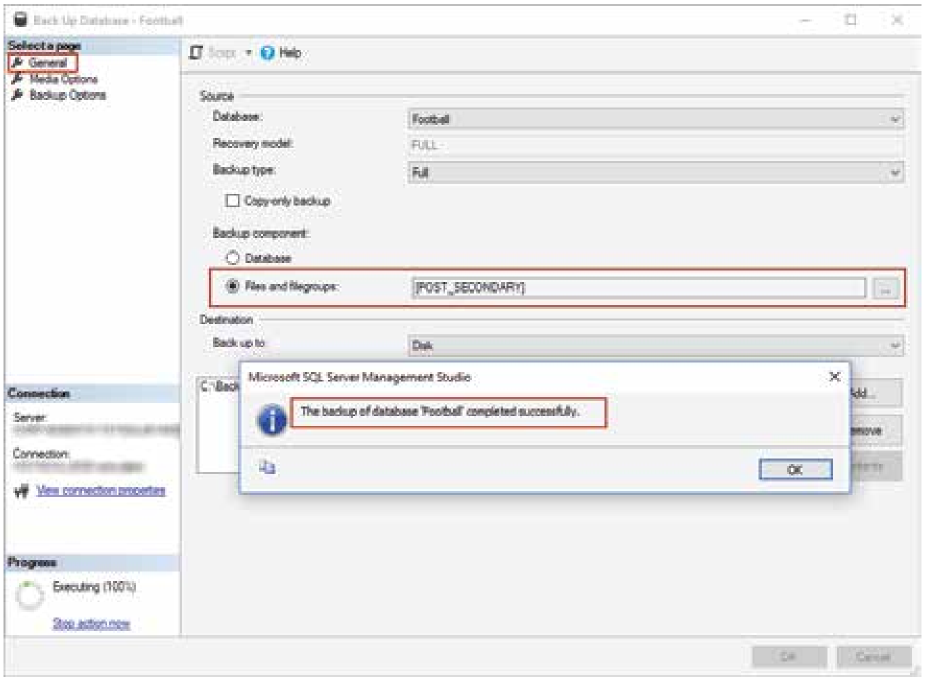
Source
Database: Football
Recovery model: FULL
Backup type: Full
☐ Copy-only backup
Backup component:
○ Database
● Files and filegroups: [POST_SECONDARY]

Destination
Back up to: Disk
C:\Backup\AdFileGroupBackMe.bak

Connection
Server: [illegible]
Connection: [illegible]
View connection properties

Progress
Ready

- Once all appropriate configurations have been completed. Click on OK to start the backup execution.
- Upon successful completion of the backup, SSMS would notify a success dialog box and also the Progress status woul be at a 100%.



Back Up Database - Football

Select a page: General, Media Options, Backup Options

Source
Database: Football
Recovery model: FULL
Backup type: Full
☐ Copy-only backup
Backup component:
○ Database
● Files and filegroups: [POST_SECONDARY]

Destination
Back up to: Disk

Microsoft SQL Server Management Studio
The backup of database 'Football' completed successfully.

Connection
Server: [illegible]
Connection: [illegible]
View connection properties

Progress
Executing (100%)
Stop action now

RESTORE FROM THE BACKUP

In order to complete a successful recovery of a SQL Server database from a failure, the set of SQL Server backups need to restore in a logically correct and meaningful sequence. The restore process supports restoring data from backup sets of a whole database, just a data file, or even a data page. Below is additional detail on the different restore types.

- The database (a complete database restore)
  - The whole database is restored and recovered, and the database is offline for the duration of the restore and recovery operations.
- The data file (a file restore)
  - A data file or a set of files is restored and recovered. During a file restore, the filegroups that contain the files are automatically offline for the duration of the restore. Any attempt to access an offline filegroup causes an error.
- The data page (a page restore)
  - Under the full recovery model or bulk-logged recovery model, you can restore individual databases. Page restores can be performed on any database, regardless of the number of filegroups.

METHODS TO RESTORE BACKUP

Like backups, SQL Server database restore can be performed either by using the SQL Server native tools or a third-party software. The demonstrations used here are the SQL Native tools (TSQL and SSMS) as that is widely used and can be easily accessible.

RESTORE ENTIRE DATABASE FROM FULL BACKUP

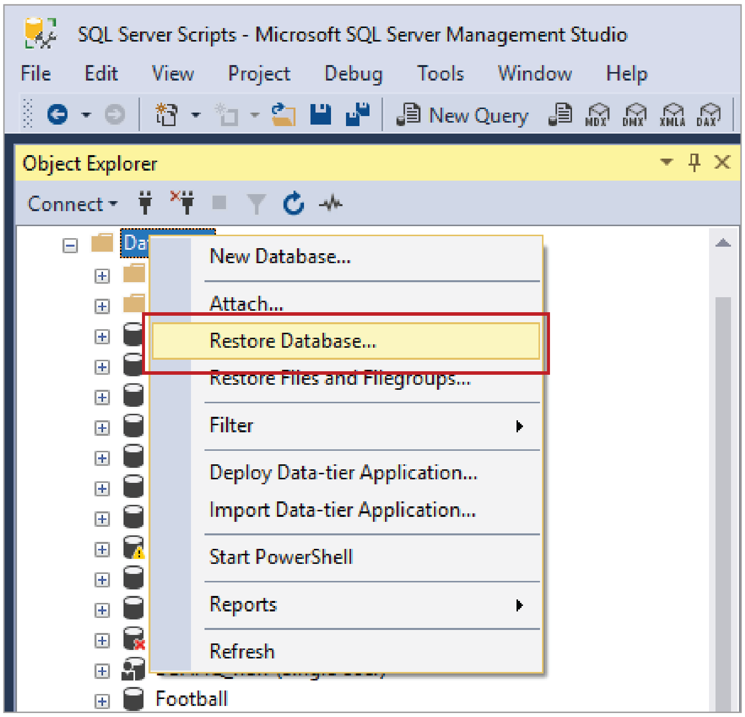
USING TSQL

Using Sample backup AdventureWorks2012 for restore demonstration.

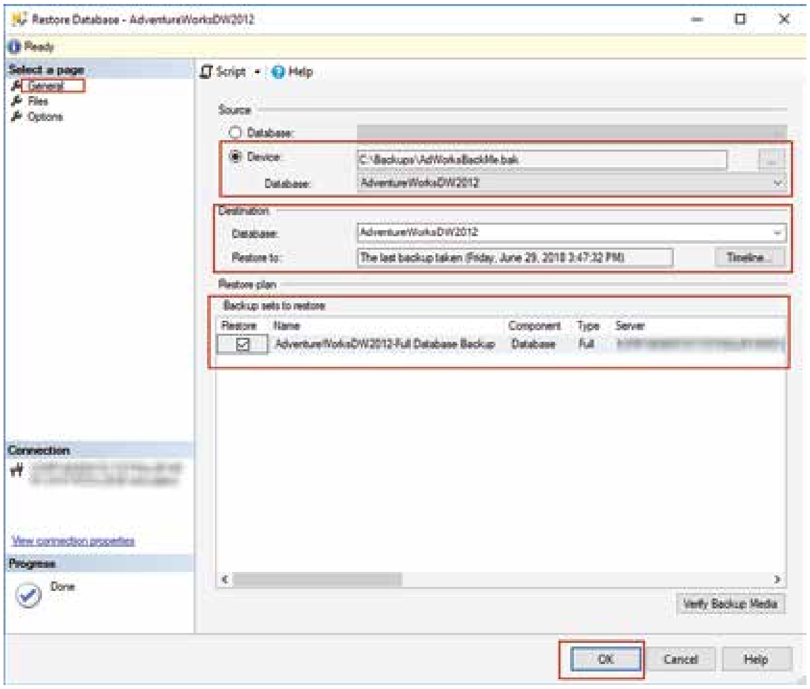
```
RESTORE DATABASE AdventureWorks2012 FROM DISK =
GO
'Z:\SQLServerBackups\AdventureWorks2012.bak'
   WITH FILE = 6 ,NORECOVERY;
RESTORE DATABASE AdventureWorks2012FROM DISK =
```

USING SSMS

- After connecting to an instance and expanding the object explorer, right click on database and select appropriate option of 'Restore Database'.



- The restore database dialog box general page provides the source and destination configurations. It also shows the restore plan available based of the backup set selected.



Source

○ Database:

● Device: C:\Backups\AdWorksBackFile.bak

Database: AdventureWorksDW2012

Destination

Database: AdventureWorksDW2012

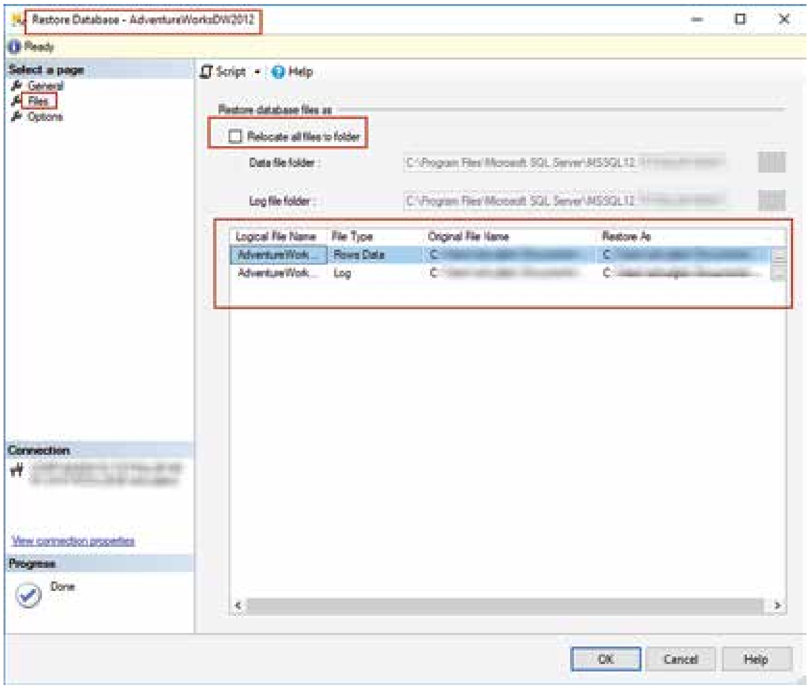
Restore to: The last backup taken (Friday, June 29, 2018 3:47:32 PM) Timeline...

Restore plan

Backup sets to restore

Restore	Name	Component	Type	Server
☑	AdventureWorksDW2012-Full Database Backup	Database	Full	[redacted]

- The 'Files' page allows you to relocate the physical database files from the default restore location to a desired location.
- Same page gives you the ability to change the logical file name and restored file names as desired.



Restore Database - AdventureWorksDW2012

Ready

Select a page
- General
- Files
- Options

Script ▾ Help

Restore database files as

☐ Relocate all files to folder

Data file folder : C:\Program Files\Microsoft SQL Server\MSSQL12[redacted]

Log file folder : C:\Program Files\Microsoft SQL Server\MSSQL12[redacted]

Logical File Name	File Type	Original File Name	Restore As
AdventureWork...	Rows Data	C[redacted]	C[redacted]
AdventureWork...	Log	C[redacted]	C[redacted]

Connection

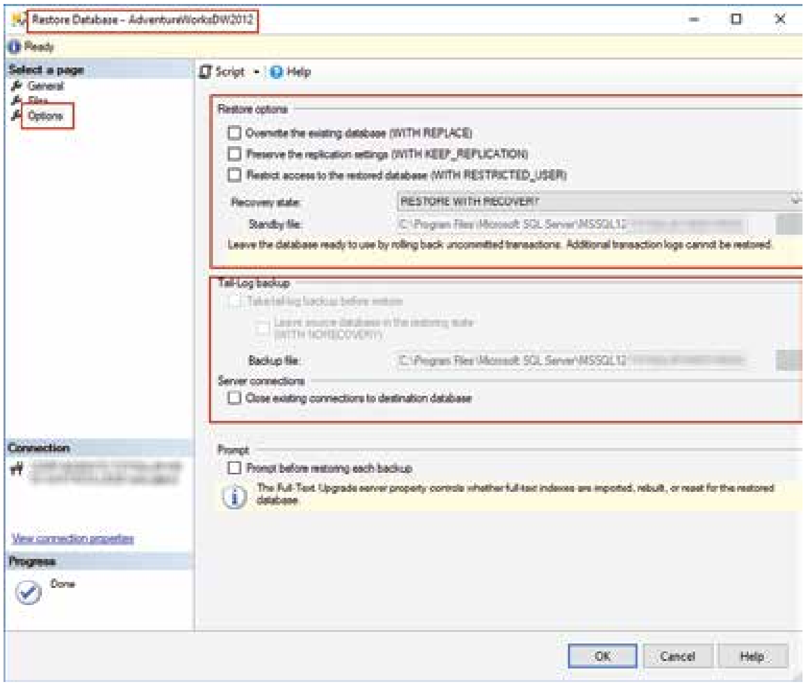
[redacted]

View connection properties

Progress

Done

- The last page 'Options' provides options of what state the data should be left in after the restore is complete. You have 3 states of:
  - o WITH RECOVERY
  - o WITH NO RECOVERY
  - o WITH STANDB
- There are the options but not required to either REPLACE, KEEP_REPLICATON or WITH RESTRICTED_USER.



Restore Database - AdventureWorksDW2012

Ready

Select a page
- General
- Files
- Options

Script ▾ Help

Restore options

☐ Overwrite the existing database (WITH REPLACE)

☐ Preserve the replication settings (WITH KEEP_REPLICATION)

☐ Restrict access to the restored database (WITH RESTRICTED_USER)

Recovery state: RESTORE WITH RECOVERY

Standby file: C:\Program Files\Microsoft SQL Server\MSSQL12[redacted]

Leave the database ready to use by rolling back uncommitted transactions. Additional transaction logs cannot be restored.

Tail-Log backup

☐ Take tail-log backup before restore

☐ Leave source database in the restoring state (WITH NORECOVERY)

Backup file: C:\Program Files\Microsoft SQL Server\MSSQL12[redacted]

Server connections

☐ Close existing connections to destination database

Prompt

☐ Prompt before restoring each backup

The Full-Text Upgrade server property controls whether full-text indexes are imported, rebuilt, or reset for the restored database.

Connection

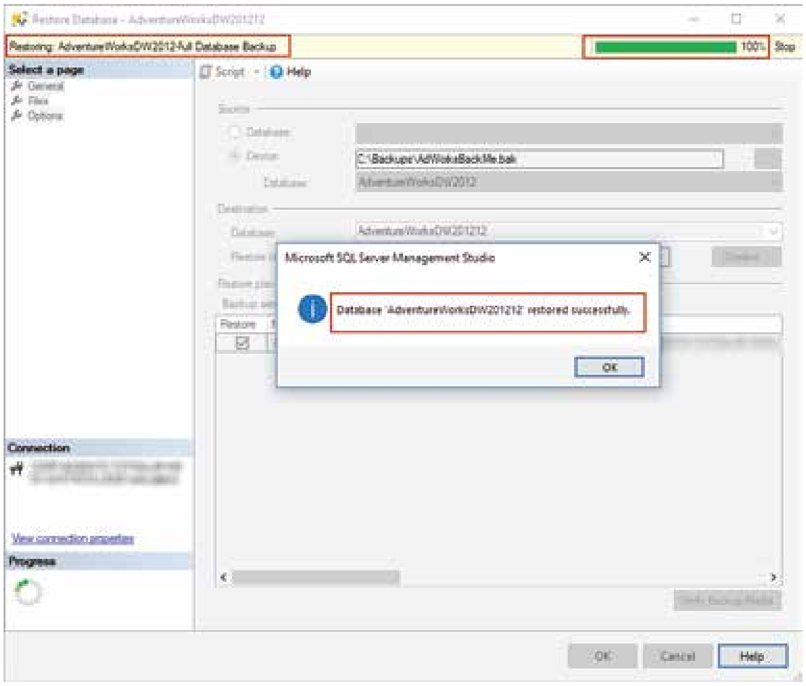
[redacted]

View connection properties

Progress

Done

- Once all appropriate restore options have been configured. You can click OK to start the restore process.
- Based on the size of the backup, number of backup files to be restored, options selected, and even performance of the server, the time for a completion of the restore will vary.



Restoring: AdventureWorksDW2012-Full Database Backup

Microsoft SQL Server Management Studio

Database 'AdventureWorksDW201212' restored successfully.

RESTORE PART OF A DATABASE (PIECEMEAL RESTORE)

This restore process is only for databases running on the Enterprise edition of SQL Server and contain multiple files or filegroups; and only for read-only filegroups if the recovery model is simple. This restore allows databases that contain multiple filegroups to be recovered in stages. It involves a series of restores starting from the primary and followed by one or more secondary filegroups. It supports all recovery models but works much better with full and bulk-logged as compared to simple. To read more on this type of restore, this link would be helpful. Due to the complexity of the sequence restore, it works with TSQL and not SSMS.

USING TSQL

Using Sample backup AdventureWorks2012 for restore demonstration

```
RESTORE DATABASE AdventureWorks2012 FILEGROUP='Primary' FROM backup1
   WITH PARTIAL, NORECOVERY
RESTORE DATABASE AdventureWorks2012 FILEGROUP='A' FROM backup2
   WITH NORECOVERY
RESTORE LOG AdventureWorks2012 FROM backup3 WITH NORECOVERY
RESTORE LOG AdventureWorks2012 FROM backup4 WITH NORECOVERY
RESTORE LOG AdventureWorks2012 FROM backup5 WITH NORECOVERY
RESTORE LOG AdventureWorks2012 FROM tailLogBackup WITH RECOVERY
```

RESTORE SPECIFIC FILES OR FILEGROUPS TO A DATABASE

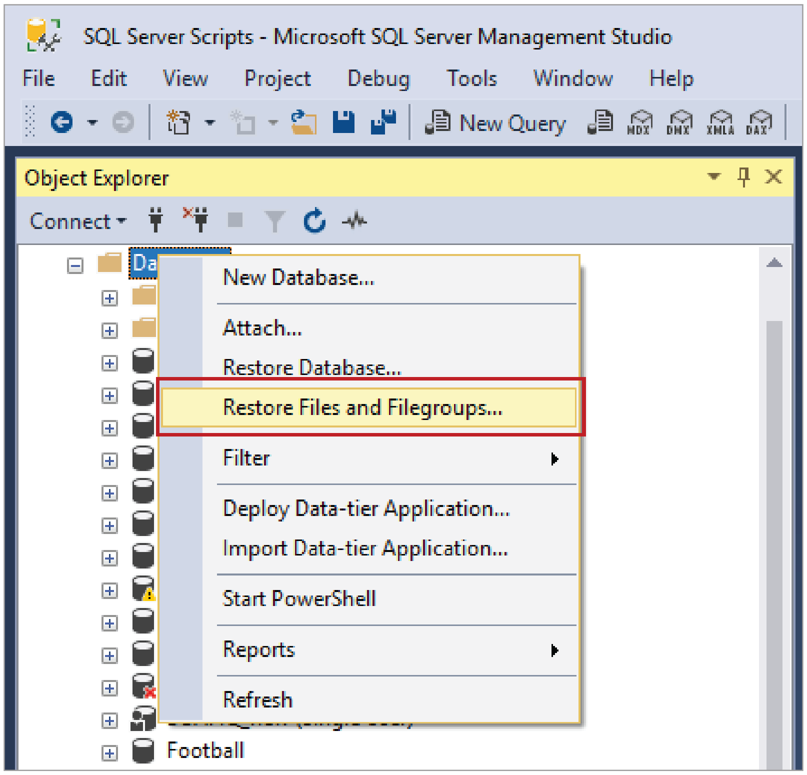
USING TSQL

Using Sample backup MyDatabase for restore demonstration.

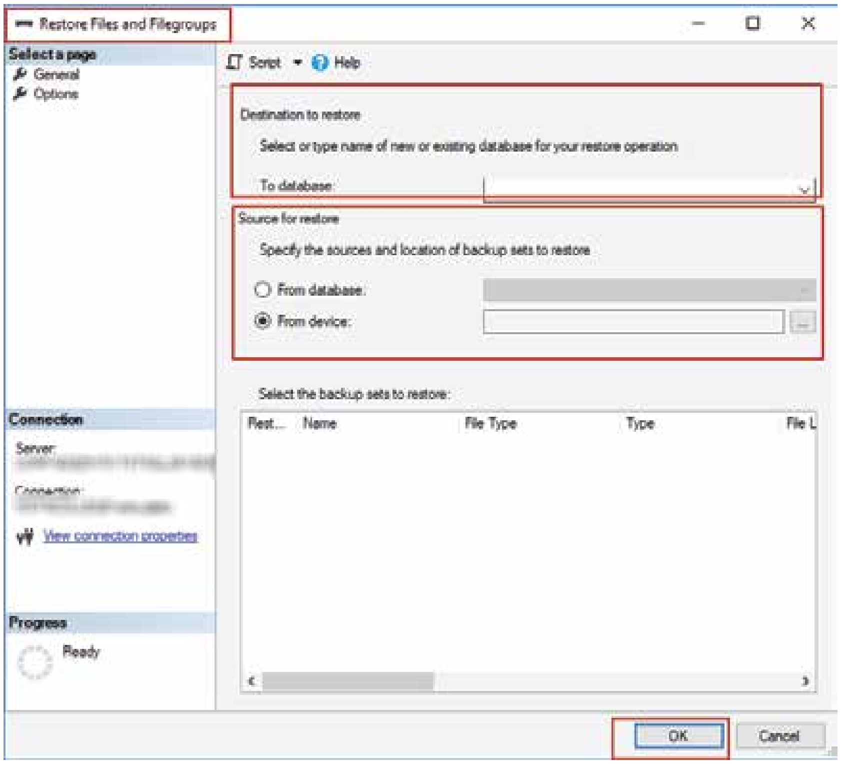
```
USE master;
GO
-- Restore the files and filesgroups for MyDatabase.
RESTORE DATABASE MyDatabase
   FILE = 'MyDatabase_data_1',
   FILEGROUP = 'new_customers',
   FILE = 'MyDatabase_data_2',
   FILEGROUP = 'first_qtr_sales'
   FROM MyDatabase_1
   WITH NORECOVERY;
GO
```

USING SSMS

- Right Click 'Databases' under Object Explorer and select 'Restore Files and Filegroups'.



- Under the General Page of the Restore Files and Filegroups dialog box, you have the destination to restore and source for restore configurations.
- Select device or database to locate your backup files or filegroups.
- Select from the drop down the database to restore to or type a new name if needed.



Destination to restore

Select or type name of new or existing database for your restore operation

To database:

Source for restore

Specify the sources and location of backup sets to restore

○ From database:

● From device:

Select the backup sets to restore:

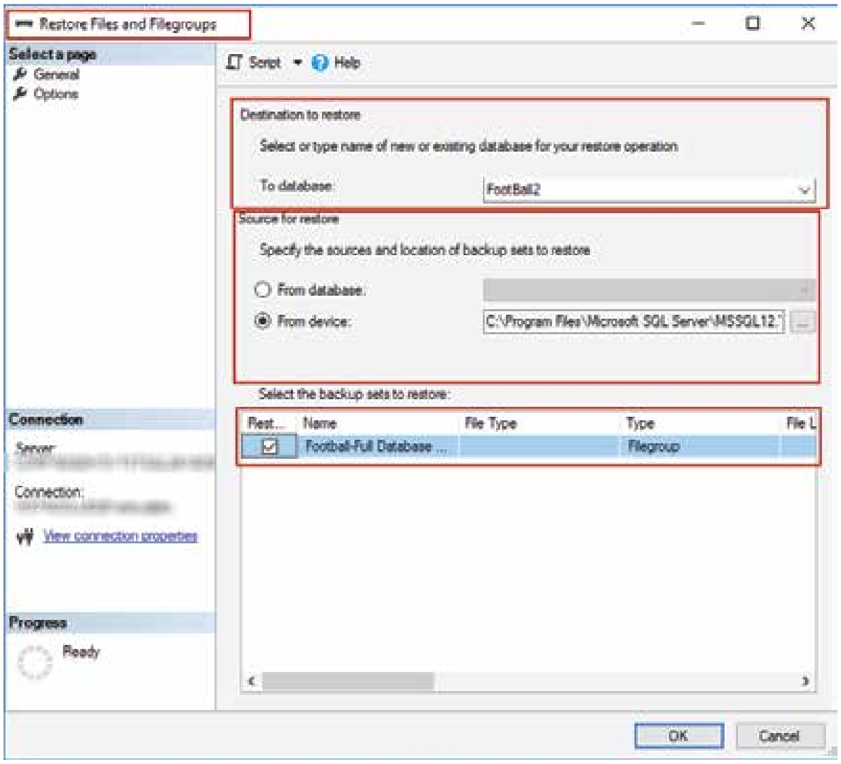
Rest...	Name	File Type	Type	File L

Server: [redacted]

Connection: [redacted]

Ready

- Below screenshot shows a sample restore configuration for filegroup backup on a device to be restored to a new database.



Destination to restore

Select or type name of new or existing database for your restore operation

To database: FootBall2

Source for restore

Specify the sources and location of backup sets to restore

○ From database:

● From device: C:\Program Files\Microsoft SQL Server\MSSQL12.

Select the backup sets to restore:

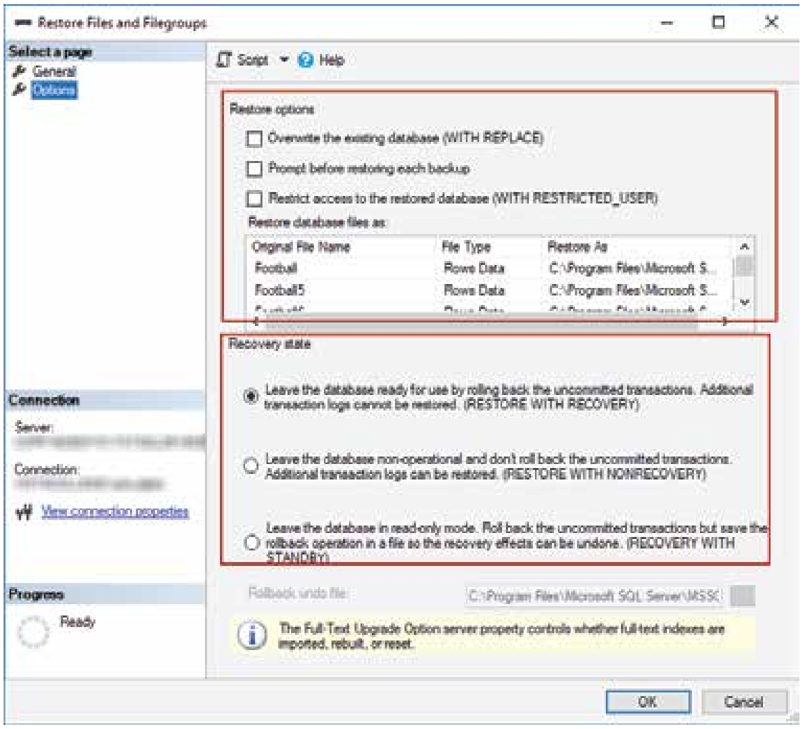
Rest...	Name	File Type	Type	File L
☑	Football-Full Database ...		Filegroup	

Server: [redacted]

Connection: [redacted]

Ready

- Under the Options page, you can optionally overwrite existing database WITH REPLACE, prompt before restoring each backup or restrict access WITH RESTRICTED_USER.
- Additionally, the required selection is setting the recovery state of the database once restore is complete.



Restore Files and Filegroups

Select a page: General, Options

Restore options

- □ Overwrite the existing database (WITH REPLACE)
- □ Prompt before restoring each backup
- □ Restrict access to the restored database (WITH RESTRICTED_USER)

Restore database files as:

Original File Name	File Type	Restore As
Football	Rows Data	C:\Program Files\Microsoft S...
Football5	Rows Data	C:\Program Files\Microsoft S...

Recovery state

- ● Leave the database ready for use by rolling back the uncommitted transactions. Additional transaction logs cannot be restored. (RESTORE WITH RECOVERY)
- ○ Leave the database non-operational and don't roll back the uncommitted transactions. Additional transaction logs can be restored. (RESTORE WITH NONRECOVERY)
- ○ Leave the database in read-only mode. Roll back the uncommitted transactions but save the rollback operation in a file so the recovery effects can be undone. (RECOVERY WITH STANDBY)

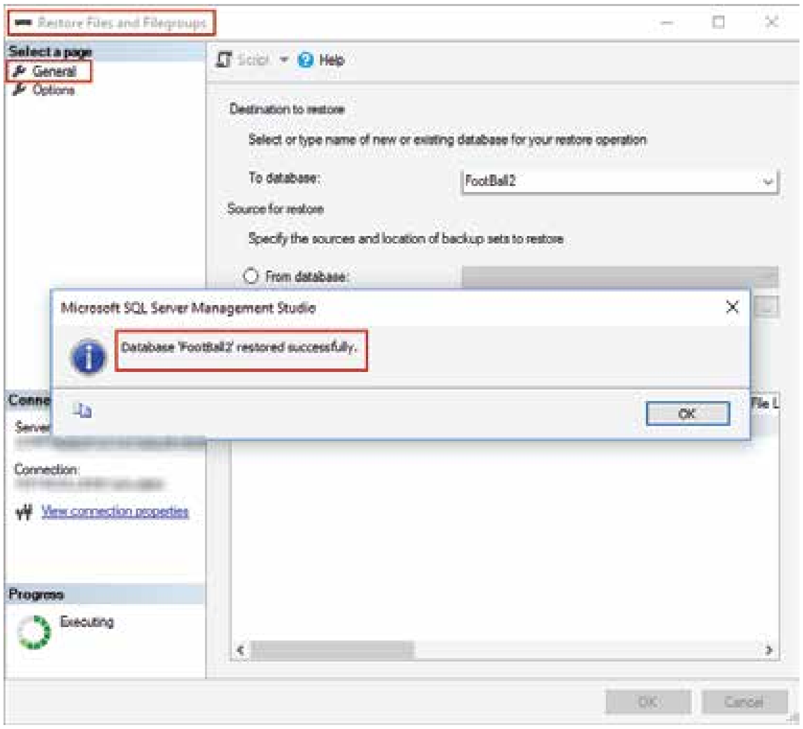
Rollback undo file: C:\Program Files\Microsoft SQL Server\MSSC

Connection — Server: [redacted] Connection: [redacted] View connection properties

Progress: Ready

The Full-Text Upgrade Option server property controls whether full-text indexes are imported, rebuilt, or reset.

- When all configurations is set right and ready to restore – click OK.
- A prompt of 'restore successful' will appear once restore is complete.



Restore Files and Filegroups

Select a page: General, Options

Destination to restore

Select or type name of new or existing database for your restore operation

To database: FootBall2

Source for restore

Specify the sources and location of backup sets to restore

○ From database:

Microsoft SQL Server Management Studio

Database 'FootBall2' restored successfully.

Connection — Server: [redacted] Connection: [redacted] View connection properties

Progress: Executing

RESTORE SPECIFIC PAGES TO A DATABASE

The idea of performing a page restore is to restore one or more damaged pages without having to restore the entire corrupt database. Pages of a database that encounter an error are usually marked as "suspect" and that information is stored in the system database MSDB. Page restore supports only FULL or BULK-LOGGED recovery model. Only database pages can be restored. The following are not supported for page restore:

- Allocation pages: Global Allocation Map (GAM) pages, Shared Global Allocation Map (SGAM) pages, and Page Free Space (PFS) pages.
- Page 0 of all data files (the file boot page)
- Page 1:9 (the database boot page)
- Full-text catalog
- Transaction log

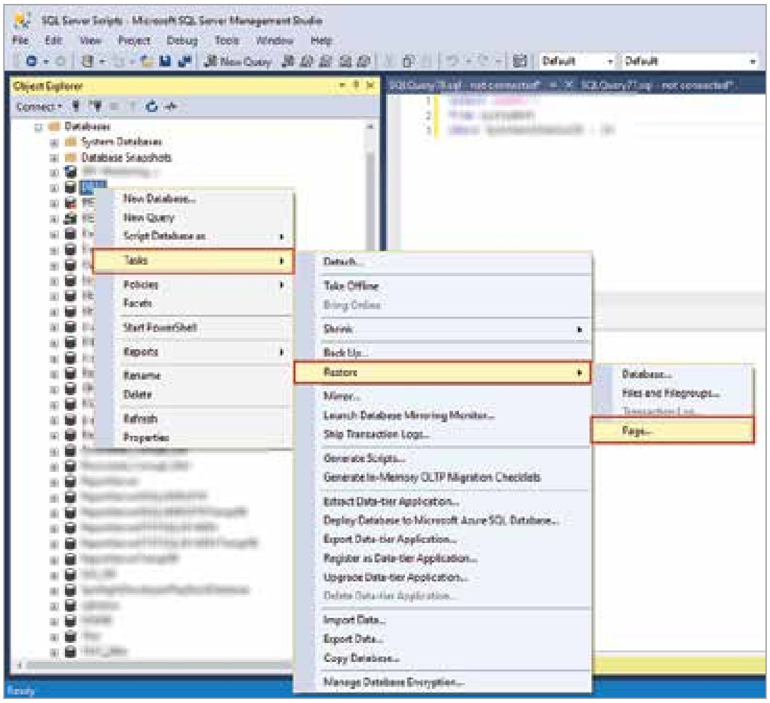
USING TSQL

Using Sample backup WideWorldImporters for restore demonstration.

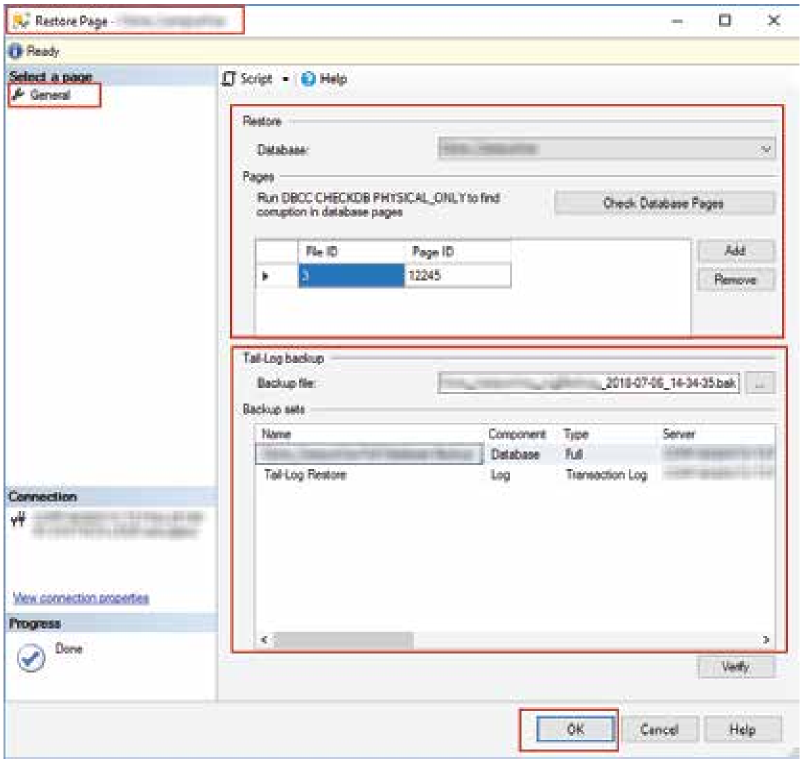
```
USE [master]
RESTORE DATABASE [WideWorldImporters]
 PAGE='3:22456'
 FROM  DISK = N'C:\Program Files\Microsoft SQL 
Server\MSSQL14.MSSQLSERVER\MSSQL\Backup\WideWorldImporters_LogBackup_2017-11-15_09-14-34.bak'
 WITH  FILE = 2,  NORECOVERY,  NOUNLOAD,  STATS = 5
```

USING SSMS

- From object explorer after a successful connection to the SQL instance, right-click the database, select task, select restore and then select page.


SQL Server Scripts - Microsoft SQL Server Management Studio
File Edit View Project Debug Tools Window Help
New Query
Default
Default
Object Explorer
Connect
Databases
System Databases
Database Snapshots
New Database...
New Query
Script Database as
Tasks
Policies
Facets
Start PowerShell
Reports
Rename
Delete
Refresh
Properties
Detach...
Take Offline
Bring Online
Shrink
Back Up...
Restore
Mirror...
Launch Database Mirroring Monitor...
Ship Transaction Logs...
Generate Scripts...
Generate In-Memory OLTP Migration Checklists
Extract Data-tier Application...
Deploy Database to Microsoft Azure SQL Database...
Export Data-tier Application...
Register as Data-tier Application...
Upgrade Data-tier Application...
Delete Data-tier Application...
Import Data...
Export Data...
Copy Database...
Manage Database Encryption...
Database...
Files and Filegroups...
Transaction Log...
Page...
Ready

- Under the General Page of Restore Page dialog box, you can add specific Page IDs and File IDs.
- The check database pages button is to check that added page information is valid to be restored.
- Specify the tail-log backup file and click ok to perform the page restore.



Restore Page - [redacted]

Ready

Select a page
- General

Script ▾ Help

Restore

Database: [redacted]

Pages

Run DBCC CHECKDB PHYSICAL_ONLY to find corruption in database pages

	File ID	Page ID
▸	1	12245

Tail-Log backup

Backup file: [redacted]_2018-07-06_14-34-35.bak

Backup sets

Name	Component	Type	Server
[redacted]	Database	Full	[redacted]
Tail-Log Restore	Log	Transaction Log	[redacted]

Connection

[redacted]

View connection properties

Progress

Done

RESTORE A TRANSACTION LOG ONTO A DATABASE

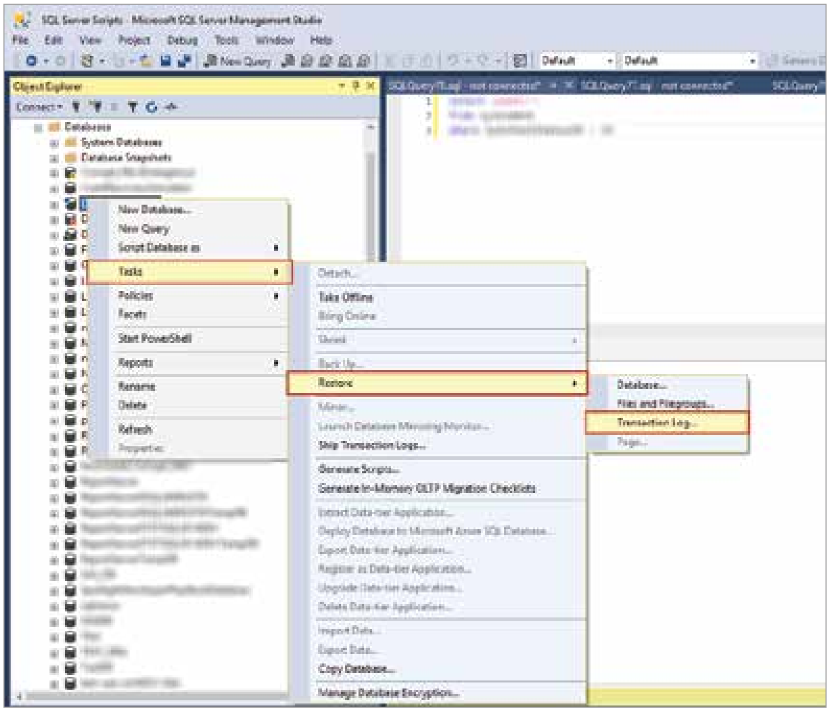
USING TSQL

Using Sample backup AdventureWorks2012 for restore demonstration.

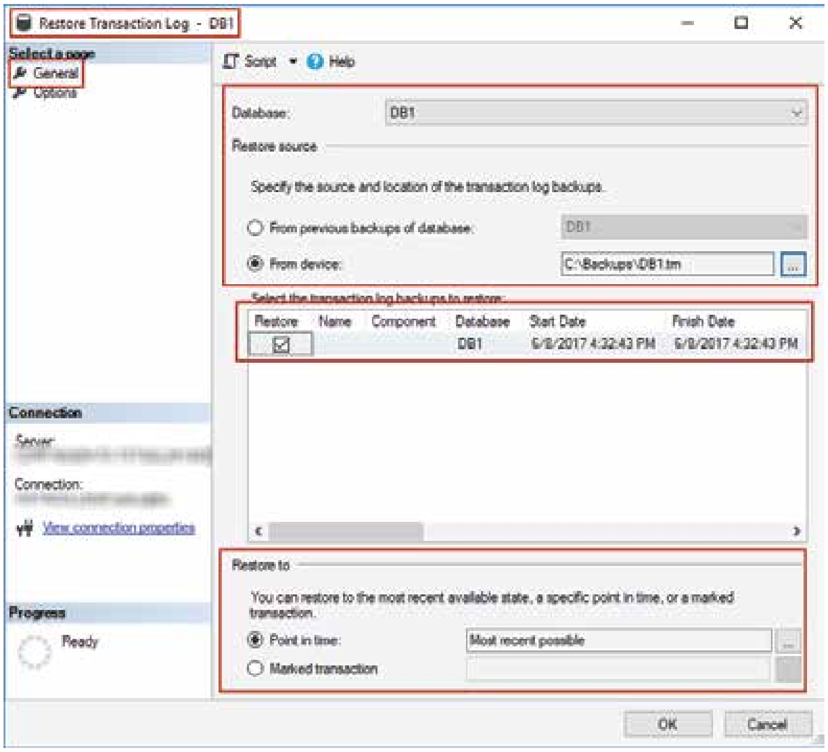
```
RESTORE DATABASE AdventureWorks2012 FROM AdventureWorks2012_1  WITH
NORECOVERY;
GO
RESTORE LOG AdventureWorks2012 FROM AdventureWorks2012_log WITH FILE =
1,NORECOVERY;
GO
```

USING SSMS

- Connect to the appropriate SQL Server instance and in Object Explorer, click the server name to expand the server tree.
- Expand Databases and either right click a user database or expand System Databases and right click the appropriate system database.
- Select Tasks, select Restore, and then click Transaction Log, which displays the Restore Transaction Log dialog box.


SQL Server Scripts - Microsoft SQL Server Management Studio
File Edit View Project Debug Tools Window Help
New Query
Default
Default
Object Explorer
Connect
Databases
System Databases
Database Snapshots
New Database...
New Query
Script Database as
Tasks
Policies
Facets
Start PowerShell
Reports
Rename
Delete
Refresh
Properties
Detach...
Take Offline
Bring Online
Shrink
Back Up...
Restore
Mirror...
Launch Database Mirroring Monitor...
Ship Transaction Logs...
Generate Scripts...
Generate In-Memory OLTP Migration Checklists
Extract Data-tier Application...
Deploy Database to Microsoft Azure SQL Database...
Export Data-tier Application...
Register as Data-tier Application...
Upgrade Data-tier Application...
Delete Data-tier Application...
Import Data...
Export Data...
Copy Database...
Manage Database Encryption...
Database...
Files and Filegroups...
Transaction Log...
Page...

- Under the General page, select the database to have the transaction log restored to. Note, only databases in a restoring state will be listed in the drop down.
- Select device to located the backed up transaction log and select a point in time restore or a specified marked transaction.



Database: DB1

Restore source

Specify the source and location of the transaction log backups.

○ From previous backups of database: DB1

● From device: C:\Backups\DB1.trn

Select the transaction log backups to restore:

Restore	Name	Component	Database	Start Date	Finish Date
☑			DB1	6/8/2017 4:32:43 PM	6/8/2017 4:32:43 PM

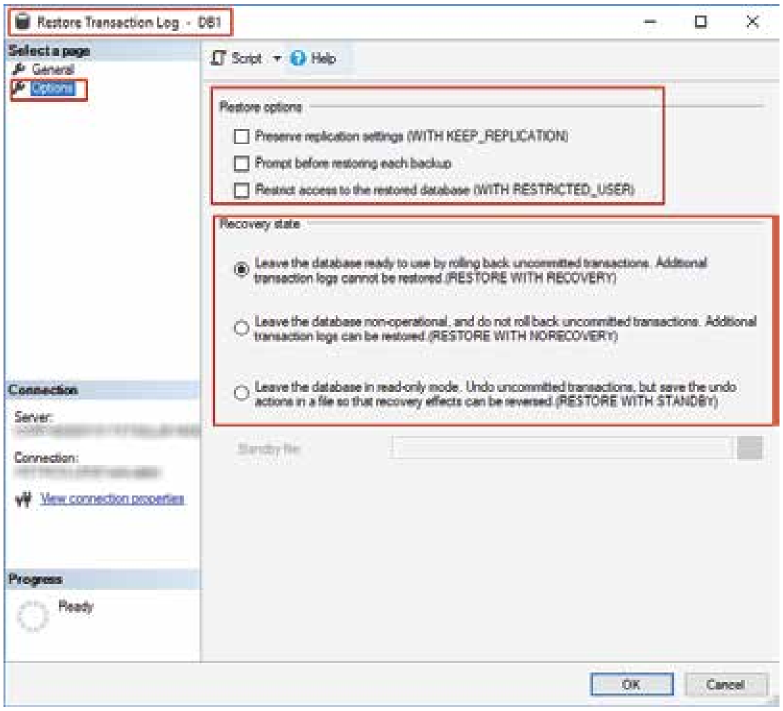
Restore to

You can restore to the most recent available state, a specific point in time, or a marked transaction.

● Point in time: Most recent possible

○ Marked transaction

- Under the Options page, you can optionally overwrite existing database WITH REPLACE, prompt before restoring each backup or restrict access WITH RESTRICTED_USER.
- Additionally selection is setting the recovery state of the database once restore is complete.



Restore options

☐ Preserve replication settings (WITH KEEP_REPLICATION)
☐ Prompt before restoring each backup
☐ Restrict access to the restored database (WITH RESTRICTED_USER)

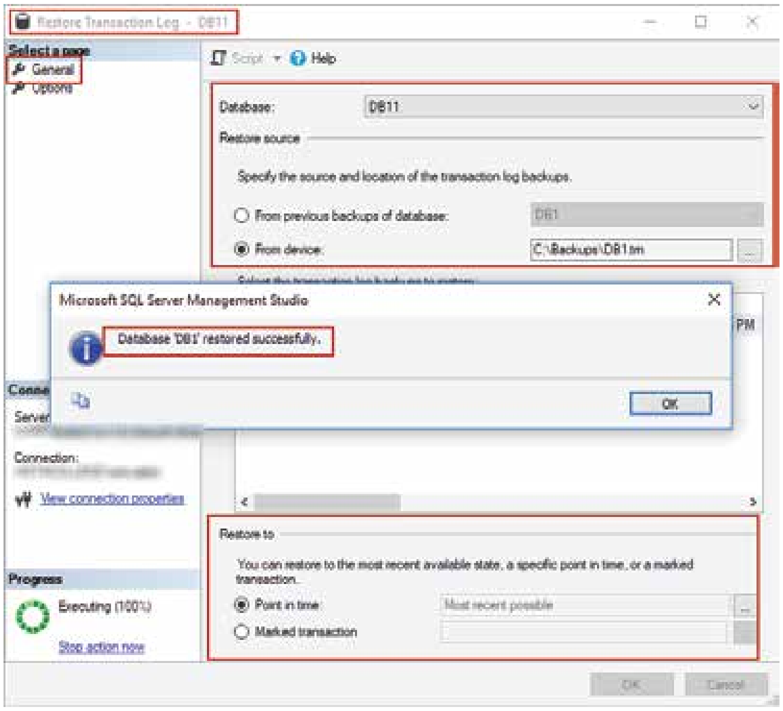
Recovery state

● Leave the database ready to use by rolling back uncommitted transactions. Additional transaction logs cannot be restored.(RESTORE WITH RECOVERY)
○ Leave the database non-operational, and do not roll back uncommitted transactions. Additional transaction logs can be restored.(RESTORE WITH NORECOVERY)
○ Leave the database in read-only mode. Undo uncommitted transactions, but save the undo actions in a file so that recovery effects can be reversed.(RESTORE WITH STANDBY)

Server: [redacted]
Connection: [redacted]

Ready

- Click OK when ready to perform restore.
- Once complete, a restore successful message box is displayed.



Database: DB11

Restore source

Specify the source and location of the transaction log backups.

○ From previous backups of database: DB1
● From device: C:\Backups\DB1.trn

Microsoft SQL Server Management Studio

Database 'DB1' restored successfully.

OK

Server: [redacted]
Connection: [redacted]

Executing (100%)

Restore to

You can restore to the most recent available state, a specific point in time, or a marked transaction.

● Point in time: Most recent possible
○ Marked transaction

REVERT A DATABASE TO A POINT IN TIME

USING TSQL

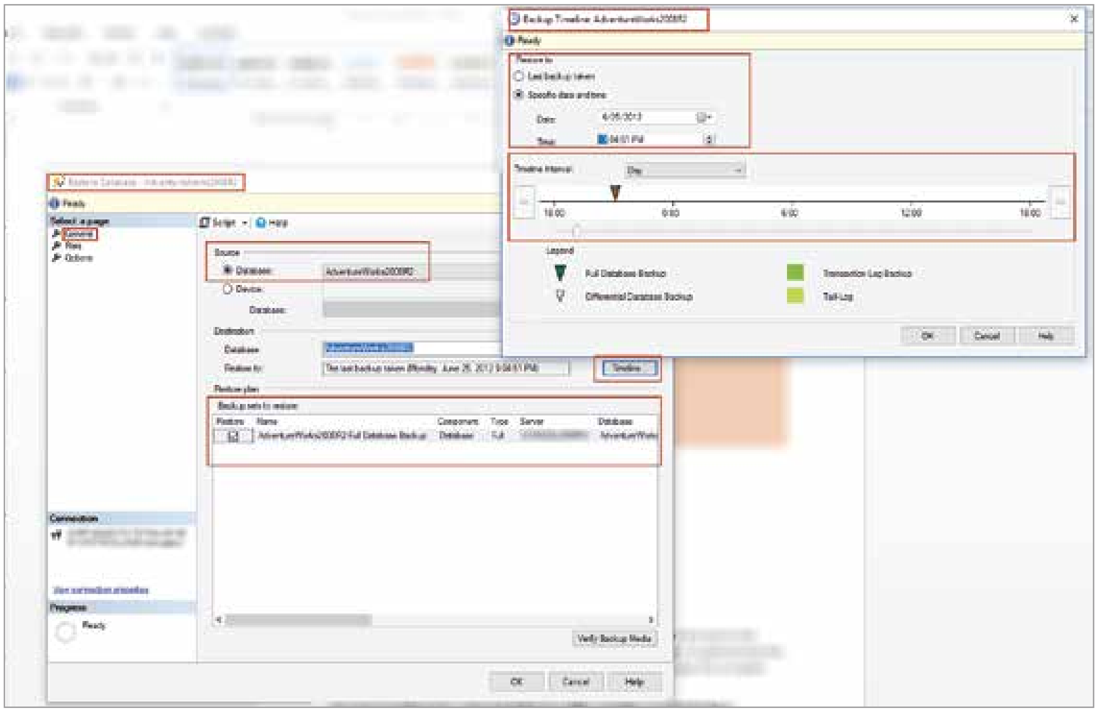
Using Sample backup AdventureWorks2012 for restore demonstration.

```
RESTORE DATABASE AdventureWorks FROM AdventureWorksBackupsWITH FILE=3,
NORECOVERY;

RESTORE LOG AdventureWorks FROM AdventureWorksBackups WITH FILE=4, NORECOVERY,
STOPAT = 'Apr 15, 2020 12:00 AM';

RESTORE LOG AdventureWorks FROM AdventureWorksBackups WITH FILE=5, NORECOVERY,
STOPAT = 'Apr 15, 2020 12:00 AM';
RESTORE DATABASE AdventureWorks WITH RECOVERY;
```

USING SSMS

- Click Timeline to access the Backup Timeline dialog box.
- In the Restore to section, click Specific date and time.
- Use either the Date and Time boxes or the slider bar to specify a specific date and time to where the restore should stop.

  Click OK.
- Click OK to start the restore to point in time.



Backup Timeline: AdventureWorks2008R2

Ready

Restore to

○ Last backup taken

● Specific date and time

Date: 6/25/2012

Time: 04:01 PM

Timeline Interval: Day

Restore Database - AdventureWorks2008R2

Source

● Database: AdventureWorks2008R2

○ Device:

Destination

Restore to: The last backup taken (Monday, June 25, 2012 9:58:51 PM)

Restore plan

Backup sets to restore:

Restore	Name	Component	Type	Server	Database
☑	AdventureWorks2008R2-Full Database Backup	Database	Full		AdventureWorks

FIXING DATABASE WITHOUT BACKUP

There are predicaments where a database ends up damaged and corrupted. Due to bad backup and recovery strategy, there are either no backups, missing sets, or corrupted backup files. In such cases, we find ourselves in the last resort option of trying to fix or repair the corrupted database.

Microsoft provided a DBCC utility that attempts to keep consistency of the database by attempting repair corrupted pages or data. There are also powerful third-party applications that can be used if all other options including DBCC CHECKDB have been utilized to no avail. We will investigate a powerful third-party SQL data recovery application later on in this paper.

For now, let's focus on the native options that SQL server provides to attempt a repair on the corrupted database.

MANUAL METHODS

Most manual methods of data recovery of a corrupted database is time consuming and the size of the database can exponentially increase the time. Another thing is there is zero guarantee that the manual process would succeed and recover the data.

REPAIR USING DBCC CHECKDB

Repairing the corrupted databases using DBCC CHECKDB checks the logical and physical integrity of all the objects in the specified database. The following operations are performed during the run of DBCC CHECKDB:

- Runs DBCC CHECKALLOC on the database.
- Runs DBCC CHECKTABLE on every table and view in the database.
- Runs DBCC CHECKCATALOG on the database.
- Validates the contents of every indexed view in the database.
- Validates link-level consistency between table metadata and file system directories and files when storing varbinry (max) data in the file system using FILESTREAM.
- Validates the Service Broker data in the database.

After a complete consistency check, a report is generated which details all information and errors encountered. In cases of consistency or allocation errors, a minimum repair level is recommended.

NOTE: A repair is not performed with a DBCC CHECKDB, you need to run the command with the appropriate repair option for that to occur. A repair is not guaranteed and might have to run multiple times and there is possible loss of data.

REPAIR FAST

This option maintains the database syntax for backward compatibility only. No repair actions are performed.

```sql
ALTER DATABASE AdventureWorks2008R2 SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
ALTER DATABASE AdventureWorks2008R2 SET EMERGENCY;
BEGIN TRANSACTION;
DBCC CHECKDB ('AdventureWorks2008R2', REPAIR_FAST) WITH ALL_ERRORMSGS;
--COMMIT
ALTER DATABASE AdventureWorks2008R2 SET MULTI_USER;
```

REPAIR RE-BUILD

To perform repairs that have no possibility of data loss. These can include quick repairs, such as repairing missing rows in non-clustered indexes, and more time-consuming repairs, such as rebuilding an index. This argument does not repair errors involving FILESTREAM data.

```sql
ALTER DATABASE AdventureWorks2008R2 SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
ALTER DATABASE AdventureWorks2008R2 SET EMERGENCY;
BEGIN TRANSACTION;
DBCC CHECKDB ('AdventureWorks2008R2', REPAIR_REBUILD) WITH ALL_ERRORMSGS;
--COMMIT
ALTER DATABASE AdventureWorks2008R2 SET MULTI_USER;
```

REPAIR ALLOW DATA LOSS

Tries to repair all reported errors. These repairs can cause some data loss. The REPAIR_ALLOW_DATA_LOSS option is a supported feature but it may not always be the best option for bringing a database to a physically consistent state. If successful, the REPAIR_ALLOW_DATA_LOSS option may result in some data loss.

```sql
ALTER DATABASE AdventureWorks2008R2 SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
ALTER DATABASE AdventureWorks2008R2 SET EMERGENCY;
BEGIN TRANSACTION;
DBCC CHECKDB ('AdventureWorks2008R2', REPAIR_ALLOW_DATA_LOSS) WITH ALL_ERRORMSGS;
--COMMIT
```

DISADVANTAGES OF THESE METHODS

- It is time-consuming and maybe end being complete waste of valuable time, should the repair run end being unsuccessful.
- The inconsistency of the database can prevent CHECKDB from looking into a part of the database and fixing that corruption. It may require running it multiple times.
- DBCC CHECKDB does not take into account foreign-key constraints.
- Some objects are un-repairable with these methods like data purity errors, PFS pages, and system table clustered index data pages.

REPAIR USING STELLAR REPAIR FOR MS SQL

OVERVIEW OF THE SOFTWARE

Stellar Repair for MS SQL software is a user-friendly application designed to repair corrupt SQL Server database primary or secondary (MDF and NDF) files. The application scans the physical database files to recover as much data as possible using complex algorithms. The software works hard to maintain the original structure and content of the corrupted database. It can achieve this while avoiding deletions or modifications to the original data.

FEATURES

REPAIRS SQL SERVER DATABASE

The Stellar Repair for MS SQL application is designed to recover data loss due to corruption in SQL Server database. The tool can handle multiple corruption scenarios of database corruption like bad I/O subsystem, SQL Server crashes, OS malfunction, unexpected system shutdown, virus attacks, etc. The tool scans the entire database file for corrupted entries and fixes the bit pattern to eliminate corruption and bring the database back to a consistent state.

ABILITY TO RESTORE DATABASE FILES (MDF AND NDF)

The primary and secondary database files hold the permanent records that have been written to the disk. Every SQL database would have at most one primary database file (.mdf) and multiple optional secondary (.ndf) files. The primary file (.mdf) holds the header and system table records and the secondary holds record information. The software can restore any of the corrupted MDF or NDF files of the database and bring it to a consistent state.

ABILITY TO RECOVER DELETED RECORDS

Aside repairing a damaged or corrupted database file(s), it also can recover deleted records of corrupt database while performing the repair function. This feature allows you to restore all your deleted records without any modifications in the original database object and save them in a newly created object. It allows further analysis to determine the difference between recovered deleted records and repaired corrupt file and save them as required.

MULTIPLE SAVING OPTIONS

Additionally, this tool is equipped with the capability of having multiple saving options that provides flexibility for the usage of the recovered database. A repaired SQL server database file can be saved as MS SQL, CSV, HTML, and XLS format at any desired location. It allows recovered data to be utilized right away even when SQL Server is not installed on server.

ALLOWS SELECTIVE RECOVERY

Stellar Repair for MS SQL software allows you to perform selective recovery of database objects. The functionality enables you to select specific database objects that must be recovered from the list and save them at default or any specified location on the computer.

ALLOWS SEARCHING THROUGH DATABASE OBJECTS

The Stellar Repair for MS SQL software provides a search capability of either 'Match case' or 'Match whole word' criteria. The ability to search items in the database after the scan is very helpful to locate database objects faster and intuitively.

RECOVERS ROW AND PAGE COMPRESSED DATA

This tool supports damaged or corrupted tables with PAGE and ROW compression. Furthermore, the tool also supports SCSU (Standard Compression Scheme for Unicode) for SQL Server 2008 R2 and higher and recovers all the data in its original format.

REQUIREMENTS

MINIMUM SYSTEM REQUIREMENTS

OPERATING SYSTEM: Windows Server 2012 / 2008 / 2003 and Windows 10 / 8 / 7 / Vista / XP

HARD DISK: At least 50 MB of free disk space

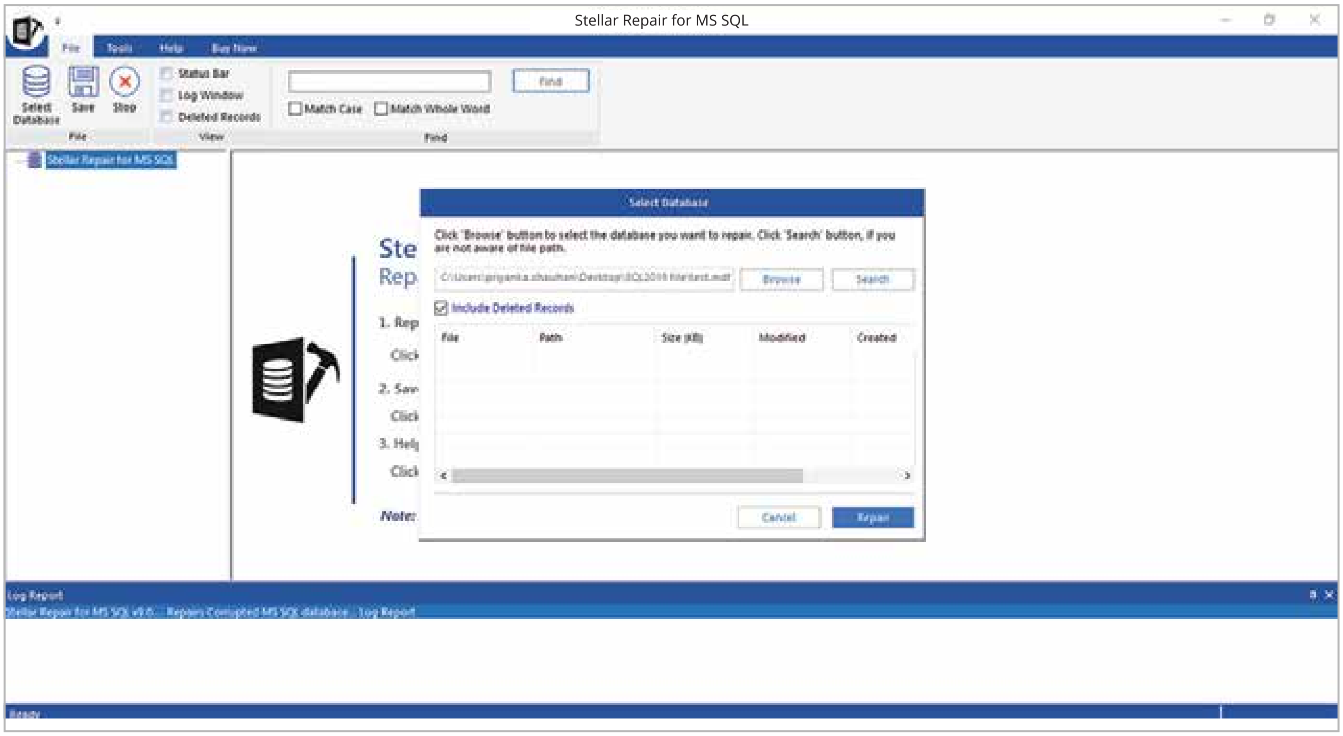
RAM: 1 GB minimum (2 GB recommended)

MS SQL SERVER: MS SQL Server 2017, MS SQL Server 2016, MS SQL Server 2014, MS SQL Server 2012, MS SQL Server 2008 R2, 2008, 2008 (64 bit), 2008 Express, MS SQL Server 2005, 2005 (64 bit), 2005 Express, MS SQL Server 2000, 2000 (64 bit), 7.0 and mixed formats.

HOW IT WORKS

The Stellar Repair for MS SQL is a very simple intuitive tool that is easy to install and use without a server reboot or additional configurations. The section below provide a quick user guide steps on how to install and repair a corrupt databases.

- First download and install the Stellar Repair for MS SQL Software from here. For testing purposes, you are provided a demo version with all features except saving the repaired .MDF file.
- On the main screen of the application, you can navigate to the location of the corrupt MDF file by clicking 'Select Database' button and choose the file from its location.
- If not sure of the file location, you can use the Find database option by looking in drives or folders as desired.

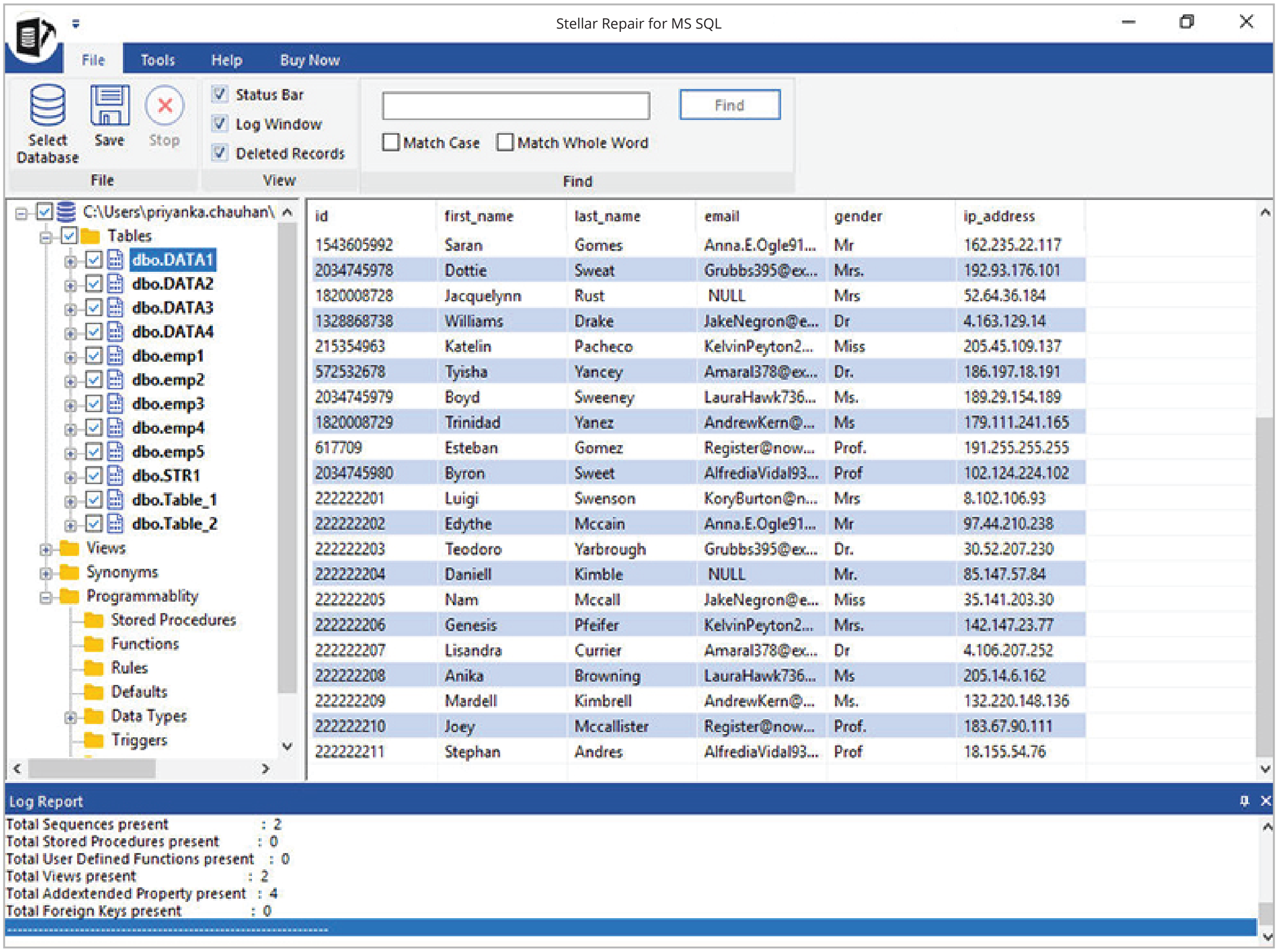


Click 'Browse' button to select the database you want to repair. Click 'Search' button, if you are not aware of file path.

C:\Users\priyanka.chauhan\Desktop\SQL2019 file\test.mdf

☑ Include Deleted Records

Another way is by checking for these options in the 'Quick Access Toolbar' or click 'Select Database' icon from 'File Menu'. Stellar Repair for MS SQL allows to include deleted records in MS SQL database (MDF) repaired file. By check-boxing the above highlighted [Include Deleted Records], the repair process will try to recover any deleted records (if found). The software is pretty fast in scanning the corrupt database file and quickly displays all the recoverable components in a tree structure. The object explorer on the left pane of the application shows the items in a sorted tree. Collapse click on any table from the tree to preview the content in the middle pane. The bottom pane shows the message log.



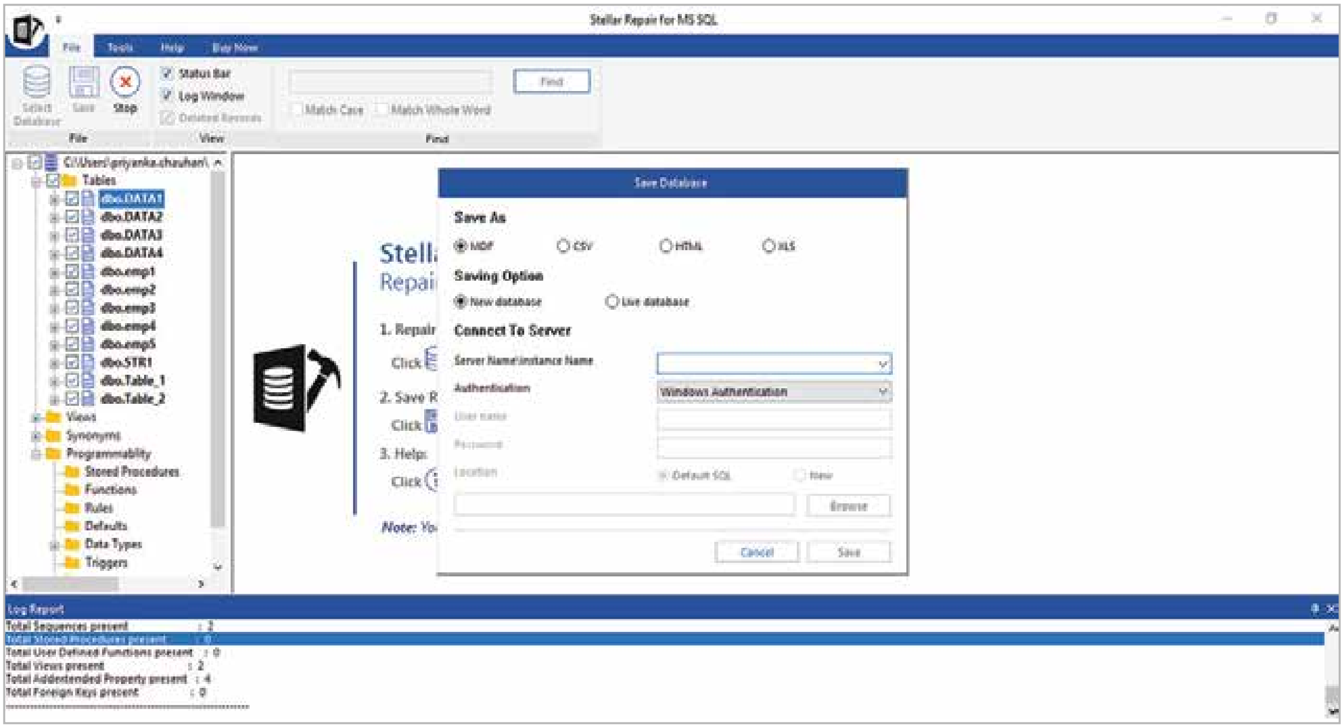
☑ Status Bar
☑ Log Window
☑ Deleted Records

id	first_name	last_name	email	gender	ip_address
1543605992	Saran	Gomes	Anna.E.Ogle91...	Mr	162.235.22.117
2034745978	Dottie	Sweat	Grubbs395@ex...	Mrs.	192.93.176.101
1820008728	Jacquelynn	Rust	NULL	Mrs	52.64.36.184
1328868738	Williams	Drake	JakeNegron@e...	Dr	4.163.129.14
215354963	Katelin	Pacheco	KelvinPeyton2...	Miss	205.45.109.137
572532678	Tyisha	Yancey	Amaral378@ex...	Dr.	186.197.18.191
2034745979	Boyd	Sweeney	LauraHawk736...	Ms.	189.29.154.189
1820008729	Trinidad	Yanez	AndrewKern@...	Ms	179.111.241.165
617709	Esteban	Gomez	Register@now...	Prof.	191.255.255.255
2034745980	Byron	Sweet	AlfrediaVidal93...	Prof	102.124.224.102
222222201	Luigi	Swenson	KoryBurton@n...	Mrs	8.102.106.93
222222202	Edythe	Mccain	Anna.E.Ogle91...	Mr	97.44.210.238
222222203	Teodoro	Yarbrough	Grubbs395@ex...	Dr.	30.52.207.230
222222204	Daniell	Kimble	NULL	Mr.	85.147.57.84
222222205	Nam	Mccall	JakeNegron@e...	Miss	35.141.203.30
222222206	Genesis	Pfeifer	KelvinPeyton2...	Mrs.	142.147.23.77
222222207	Lisandra	Currier	Amaral378@ex...	Dr	4.106.207.252
222222208	Anika	Browning	LauraHawk736...	Ms	205.14.6.162
222222209	Mardell	Kimbrell	AndrewKern@...	Ms.	132.220.148.136
222222210	Joey	Mccallister	Register@now...	Prof.	183.67.90.111
222222211	Stephan	Andres	AlfrediaVidal93...	Prof	18.155.54.76

```
Total Sequences present              : 2
Total Stored Procedures present      : 0
Total User Defined Functions present  : 0
Total Views present                  : 2
Total Addextended Property present   : 4
Total Foreign Keys present           : 0
-------------------------------------------------------------
```

From the application menu toolbar, you can search for database objects like tables with the [Find Tree Item]. You can 'Match Case' or 'Match Whole Words'. Either way you can sift through an unfamiliar database or a large database with ease. After the repair, if you wish to save the database – select the database objects to be recovered from the tree and click the 'Save' button from the tool bar. Alternately, you can also click the 'Save' button from the Quick Access Toolbar. You can save just the specific objects or the entire database.

The 'Save' button another window opens for connecting to the server or exporting the data to other mediums. If you select MSSQL and click ok, then you are prompted with another dialog box to connect to the destination SQL Server. You will also have the other for creating a new database or [LIVE] database. The repaired database files can be saved at the default SQL Location for the target server or you can select a new location.



Save Database

Save As

● MDF ○ CSV ○ HTML ○ XLS

Saving Option

● New database ○ Live database

Connect To Server

Server Name\Instance Name:

Authentication: Windows Authentication

User name:

Password:

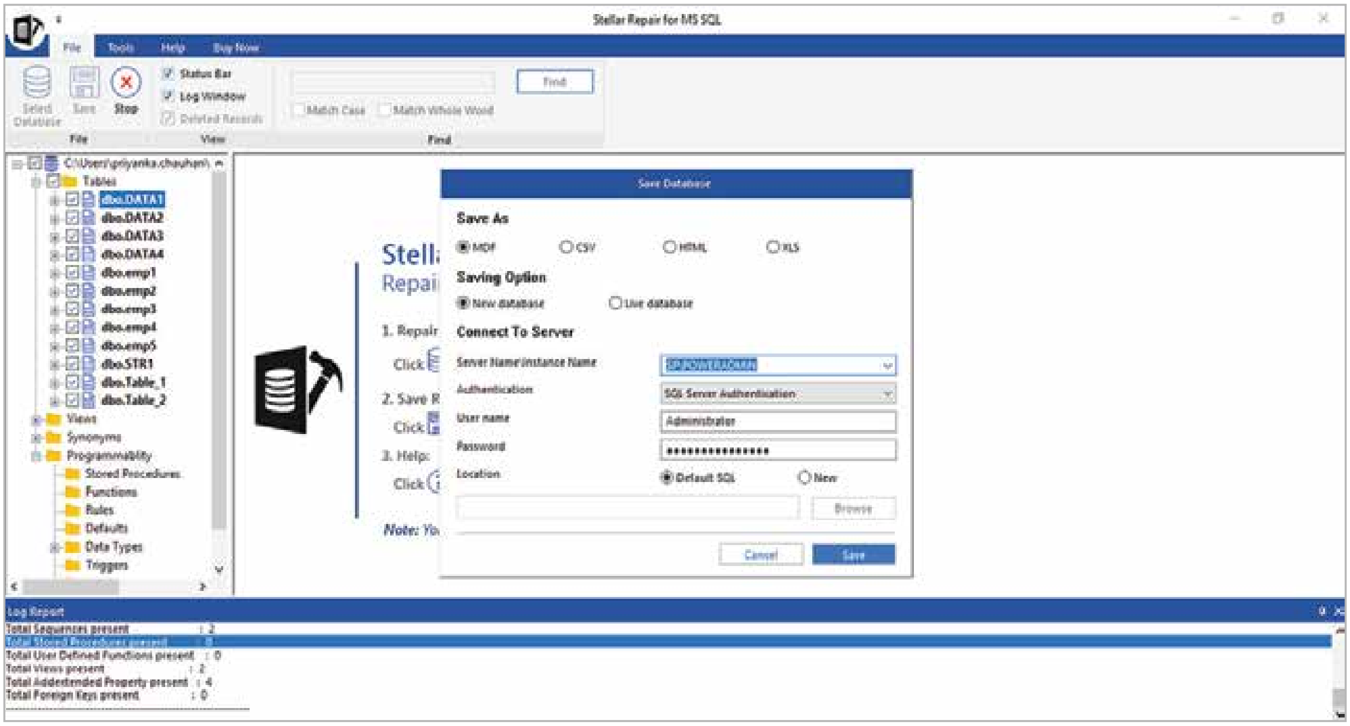
Location: ● Default SQL ○ New

Browse

Log Report

```
Total Sequences present              : 2
Total Stored Procedures present      : 0
Total User Defined Functions present : 0
Total Views present                  : 2
Total Addentended Property present   : 4
Total Foreign Keys present           : 0
```

After connection is established and database is successfully saved, a message "File saved at the desired path" is displayed for confirmation.



Save Database

Save As

● MDF ○ CSV ○ HTML ○ XLS

Saving Option

● New database ○ Live database

Connect To Server

Server Name\Instance Name: SP.POWERADMIN

Authentication: SQL Server Authentication

User name: Administrator

Password: ****************

Location: ● Default SQL ○ New

Browse

Log Report

```
Total Sequences present              : 2
Total Stored Procedures present      : 0
Total User Defined Functions present : 0
Total Views present                  : 2
Total Addentended Property present   : 4
Total Foreign Keys present           : 0
```

CONCLUSION

By now you would have realized the importance of database backups and how database recovery can end up becoming a major problem for any business in times of data loss. In order to have a good backup strategy, you must understand the applications and business SLAs to ensure that you're maintaining the right level of data recovery.

It is important to test the restores as often as you can, especially if your backups are complex. Administrating your backups can be a very simple process, but it's pertinent that you get it right all the time. The worst is when you have a weak backup strategy and in times of need, these database backups tend to be useless. Remember that all is not lost when you find yourself in the worst-case scenario. Look into the powerful third-party data recovery tool – Stellar Repair for MS SQL tool to help you repair and recover data from your corrupt database. Send a mail or call-in to reach out to the Stellar Team to enquire about your go-to-product for resolving database corruption.

REFERENCES

"Restore a SQL Server Database to a Point in Time (Full Recovery Model)." Microsoft Docs, docs.microsoft.com/en-us/sql/relational-databases/backup-restore/restore-a-sql-server-database-to-a-point-in-time-full-recovery-model?view=sql-server-2017.

"Full Database Backups (SQL Server)." Microsoft Docs, docs.microsoft.com/en-us/sql/relational-databases/backup-restore/full-database-backups-sql-server?view=sql-server-2017.

CONTACT DETAILS

Stellar Data Recovery Inc.

48 Bridge Street Metuchen,
New Jersey 08840
United States
call us - 1-877-778-6087
email - support[at]stellarinfo.com

Stellar Information Technology Pvt. Ltd.

D16, Sector-33,
Infocity Phase II,
Gurugram-122001
call us - (0124) 432 6700
email - helpdeskdrs[at]stellarinfo.com

USA Helpline

Sales Queries (Tollfree)+1-877-778-6087

Support Queries+1-732-584-2700

UK (Europe) Helpline

Sales & Support Queries+44-203-026-5337

World Wide Helpline

Sales & Support Queries+91-124-4326-777

Australia & Asia Pacific

Sales & Support Queries+61-280149899

Netherlands

Sales & Support Queries+31-208-111-188