

Scheduling A Backup In SQL

Last Modified on 04/26/2022 8:54 am EDT

Procedure Steps

To back up an SQL Server database on a time schedule for an SQL Server 2017 Express system:

1) Create a folder named SQLBackup in the C:\ drive.

2) Copy the following code, and then start Notepad. Paste the text into Notepad.

```
=====

DECLARE @BackupFile varchar(255), @DB varchar(30), @Description varchar(255), @LogFile
varchar(50)

DECLARE @Name varchar(30), @MediaName varchar(30), @BackupDirectory nvarchar(200)

SET @BackupDirectory = 'C:\SQLBackup\'

-- Add a list of all databases you don't want to backup to this, eg: tempdb, model, Northwind,
master, msdb.

DECLARE Database_CURSOR CURSOR FOR SELECT name FROM sysdatabases WHERE name <>
'tempdb' AND name <> 'model' AND name <> 'Northwind' AND name <> 'master' AND name <>
'msdb'

OPEN Database_Cursor

FETCH next FROM Database_CURSOR INTO @DB

WHILE @@fetch_status = 0

BEGIN

    SET @Name = @DB + '( Daily BACKUP )'

    SET @MediaName = @DB + '_Dump' + CONVERT(varchar, CURRENT_TIMESTAMP , 112)

    SET @BackupFile = @BackupDirectory + CONVERT(varchar, CURRENT_TIMESTAMP , 112) + '_' +
@DB + '_' + 'Full' + '_' + '.bak'

    SET @Description = 'Normal' + ' BACKUP at ' + CONVERT(varchar, CURRENT_TIMESTAMP) + '!'


```

```

IF (SELECT COUNT(*) FROM msdb.dbo.backupset WHERE database_name = @DB) > 0 OR @DB
= 'master'

BEGIN

SET @BackupFile = @BackupDirectory + CONVERT(varchar, CURRENT_TIMESTAMP , 112) + '_' +
@DB + '_' + 'Full' + '_' + '.bak'

-- SET some more pretty stuff for sql server.

SET @Description = 'Full' + ' BACKUP at ' + CONVERT(varchar, CURRENT_TIMESTAMP) + '!'

END

ELSE

BEGIN

SET @BackupFile = @BackupDirectory + @DB + '_' + 'Full' + '_' +
CONVERT(varchar, CURRENT_TIMESTAMP , 112) + '.bak'

-- SET some more pretty stuff for sql server.

SET @Description = 'Full' + ' BACKUP at ' + CONVERT(varchar, CURRENT_TIMESTAMP) + '!'

END

BACKUP DATABASE @DB TO DISK = @BackupFile

WITH NAME = @Name, DESCRIPTION = @Description ,

MEDIANAME = @MediaName, MEDIADESCRIPTION = @Description ,

STATS = 10

FETCH next FROM Database_CURSOR INTO @DB

END

CLOSE Database_Cursor

DEALLOCATE Database_Cursor

=====

```

- 3) From the File menu, select Save.
- 4) In the Save as type drop-down, select "All Files".
- 5) In the File name field type backup.sql. Save this file to the C:\SQLBackup folder.
- 6) Copy the following code, and then paste the text into Notepad.

=====
`sqlcmd -S . -i "C:\SQLBackup\Backup.sql"`
=====

- 7) From the File menu, select Save.
 - 8) In the Save as type drop-down, select "All Files".
 - 9) In the File name field type backup.bat. Save this file to the C:\SQLBackup folder.
 - 10) Open Schedule Tasks in Windows (Click the Start button, then select All Programs > Accessories > System Tools > Scheduled Tasks).
 - 11) Double-click Add Scheduled Task. The Scheduled Task Wizard opens.
 - 12) Click [Next] on the first window.
 - 13) On the window where you select the program you want Windows to run, click [Browse]. Navigate to the C:\SQLBackup\backup.bat file, and then click [Open].
 - 14) Type a name for the task, select how frequently to perform the task, and then click [Next].
 - 15) Select the time and day you want the task to start, along with the day(s) of the week to run the task. Click [Next].
 - 16) Enter the Windows login (which is required), and then click [Next].
- Note: The login must have access to SQL
- 17) Click [Finish].
 - 18) To test the schedule, right-click on the schedule in Scheduled Tasks and select Run. If the backup runs successfully, a Date_AccessControl.bak file will be created in the C:\SQLBackup folder.

Applies To

OnGuard 8.0

Additional Information
