How to schedule and automate backups of SQL Server databases in SQL Server Express

You have to follow these three steps to back up your SQL Server databases by using Windows Task Scheduler:

Step A: Use SQL Server Management Studio Express or Sqlcmd to create the following stored procedure in your master database:

```
// Copyright © Microsoft Corporation.  All Rights Reserved.
// This code released under the terms of the
// Microsoft Public License (MS-PL, http://opensource.org/licenses/ms-pl.html.)
USE [master]
GO
/****** Object:  StoredProcedure [dbo].[sp_BackupDatabases] ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

-- =============================================
-- Author: Microsoft
-- Create date: 2010-02-06
-- Description: Backup Databases for SQLExpress
-- Parameter1: databaseName
-- Parameter2: backupType F=full, D=differential, L=log
-- Parameter3: backup file location
-- =============================================

CREATE PROCEDURE [dbo].[sp_BackupDatabases]
        @databaseName sysname = null,
        @backupType CHAR(1),
        @backupLocation nvarchar(200)
AS

    SET NOCOUNT ON;

        DECLARE @DBs TABLE
        (
```

```
        ID int IDENTITY PRIMARY KEY,
        DBNAME nvarchar(500)
    )

    -- Pick out only databases which are online in case ALL databases are chosen to be backed up
    -- If specific database is chosen to be backed up only pick that out from @DBs
    INSERT INTO @DBs (DBNAME)
    SELECT Name FROM master.sys.databases
    where state=0
    AND name=@DatabaseName
    OR @DatabaseName IS NULL
    ORDER BY Name

    -- Filter out databases which do not need to backed up
    IF @backupType='F'
        BEGIN
        DELETE @DBs where DBNAME IN ('tempdb','Northwind','pubs','AdventureWorks')
        END
    ELSE IF @backupType='D'
        BEGIN
        DELETE @DBs where DBNAME IN ('tempdb','Northwind','pubs','master','AdventureWorks')
        END
    ELSE IF @backupType='L'
        BEGIN
        DELETE @DBs where DBNAME IN ('tempdb','Northwind','pubs','master','AdventureWorks')
        END
    ELSE
        BEGIN
        RETURN
        END

    -- Declare variables
    DECLARE @BackupName varchar(100)
    DECLARE @BackupFile varchar(100)
    DECLARE @DBNAME varchar(300)
    DECLARE @sqlCommand NVARCHAR(1000)
DECLARE @dateTime NVARCHAR(20)
    DECLARE @Loop int

    -- Loop through the databases one by one
    SELECT @Loop = min(ID) FROM @DBs
```

```
    WHILE @Loop IS NOT NULL
    BEGIN

-- Database Names have to be in [dbname] format since some have - or _ in their name
    SET @DBNAME = '['+(SELECT DBNAME FROM @DBs WHERE ID = @Loop)+']'

-- Set the current date and time n yyyyhhmmss format
    SET @dateTime = REPLACE(CONVERT(VARCHAR, GETDATE(),106),'/','') + '_' +
REPLACE(CONVERT(VARCHAR, GETDATE(),108),':','')

-- Create backup filename in path\filename.extension format for full,diff and log backups
    IF @backupType = 'F'
        SET @BackupFile = @backupLocation+REPLACE(REPLACE(@DBNAME, '[',''),']','')+ '_FULL_'+
@dateTime+ '.BAK'
    ELSE IF @backupType = 'D'
        SET @BackupFile = @backupLocation+REPLACE(REPLACE(@DBNAME, '[',''),']','')+ '_DIFF_'+
@dateTime+ '.BAK'
    ELSE IF @backupType = 'L'
        SET @BackupFile = @backupLocation+REPLACE(REPLACE(@DBNAME, '[',''),']','')+ '_LOG_'+
@dateTime+ '.TRN'

-- Provide the backup a name for storing in the media
    IF @backupType = 'F'
        SET @BackupName = REPLACE(REPLACE(@DBNAME,'[',''),']','') +' full backup for '+ @dateTime
    IF @backupType = 'D'
        SET @BackupName = REPLACE(REPLACE(@DBNAME,'[',''),']','') +' differential backup for '+
@dateTime
    IF @backupType = 'L'
        SET @BackupName = REPLACE(REPLACE(@DBNAME,'[',''),']','') +' log backup for '+ @dateTime

-- Generate the dynamic SQL command to be executed

     IF @backupType = 'F'
            BEGIN
           SET @sqlCommand = 'BACKUP DATABASE ' +@DBNAME+  ' TO DISK = '''+@BackupFile+ ''' WITH
INIT, NAME= ''' +@BackupName+''', NOSKIP, NOFORMAT'
            END
     IF @backupType = 'D'
            BEGIN
          SET @sqlCommand = 'BACKUP DATABASE ' +@DBNAME+  ' TO DISK = '''+@BackupFile+ ''' WITH
DIFFERENTIAL, INIT, NAME= ''' +@BackupName+''', NOSKIP, NOFORMAT'
            END
```

```
    IF @backupType = 'L'
            BEGIN
          SET @sqlCommand = 'BACKUP LOG ' +@DBNAME+  ' TO DISK = '''+@BackupFile+ ''' WITH INIT,
NAME= ''' +@BackupName+''', NOSKIP, NOFORMAT'
            END

-- Execute the generated SQL command
    EXEC(@sqlCommand)

-- Goto the next database
SELECT @Loop = min(ID) FROM @DBs where ID>@Loop

END
```

Step B: In a text editor, create a batch file that is named Sqlbackup.bat, and then copy the text from one of the following examples into that file, depending on your scenario:

Example1: Full backups of all databases in the local named instance of SQLEXPRESS by using Windows Authentication

```
// Sqlbackup.bat
sqlcmd -S .\EXPRESS –E -Q "EXEC sp_BackupDatabases @backupLocation='D:\SQLBackups\',
@backupType='F'"
```

Example2: Differential backups of all databases in the local named instance of SQLEXPRESS by using a SQLLogin and its password

```
// Sqlbackup.bat
sqlcmd -U SQLLogin -P password -S .\SQLEXPRESS -Q "EXEC sp_BackupDatabases  @backupLocation
='D:\SQLBackups', @BackupType='D'"
```

Note: The SQLLogin shouldhave at least the Backup Operator role in SQL Server.

Example 3: Log backups of all databases in local named instance of SQLEXPRESS by using Windows Authentication

```
// Sqlbackup.bat
sqlcmd -S .\SQLEXPRESS -E -Q "EXEC sp_BackupDatabases
@backupLocation='D:\SQLBackups\',@backupType='L'"
```

Example 4: Full backups of the database USERDB in the local named instance of SQLEXPRESS by using Windows Authentication

```
// Sqlbackup.bat
sqlcmd -S .\SQLEXPRESS -E -Q "EXEC sp_BackupDatabases @backupLocation='D:\SQLBackups\',
@databaseName='USERDB', @backupType='F'"
```

Similarly, you can make a differential backup of USERDB by pasting in 'D' for the @backupType parameter and a log backup of USERDB by pasting in 'L' for the @backupType parameter.

Step C: Schedule a job by using Windows Task Scheduler to execute the batch file that you created in step B. To do this, follow these steps:

1. On the computer that is running SQL Server Express, click Start, point to All Programs, point to Accessories, point to System Tools, and then click Scheduled Tasks.
2. Double-click Add Scheduled Task.
3. In the Scheduled Task Wizard, click Next.
4. Click Browse, click the batch file that you created in step B, and then click Open.
5. Type SQLBACKUP for the name of the task, click Daily, and then click Next.
6. Specify information for a schedule to run the task. (We recommend that you run this task at least one time every day.) Then, click Next.
7. In the Enter the user name field, type a user name, and then type a password in the Enter the password field.

Note This user should at least be assigned the BackupOperator role at SQL Server level if you are using one of the batch files in example 1, 3, or 4.

8. Click Next, and then click Finish.
9. Execute the scheduled task at least one time to make sure that the backup is created successfully.

Note The folder for the SQLCMD executable is generally in the Path variables for the server after SQL Server is installed, but if the Path variable does not list this folder, you can find it under <Install location>\90\Tools\Binn (For example: C:\Program Files\Microsoft SQL Server\90\Tools\Binn).

Be aware of the following when you use the procedure that is documented in this article:

- The Windows Task Scheduler service must be running at the time that the job is scheduled to run. We recommend that you set the startup type for this service as Automatic. This makes sure that the service will be running even on a restart.

- There should be lots of space on the drive to which the backups are being written. We recommend that you clean the old files in the backup folder regularly to make sure that you do not run out of disk space. The script does not contain the logic to clean up old files.

Article taken from below link:
https://support.microsoft.com/en-in/help/2019698/how-to-schedule-and-automate-backups-of-sql-server-databases-in-sql-se