USE [msdb]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[uspGetPermissionsOfAllLogins\_DBsOnColumns]

AS

SET NOCOUNT ON

;

BEGIN TRY

IF EXISTS

(

SELECT \* FROM tempdb.dbo.sysobjects

WHERE id = object\_id(N'[tempdb].dbo.[#permission]')

)

DROP TABLE #permission

;

IF EXISTS

(

SELECT \* FROM tempdb.dbo.sysobjects

WHERE id = object\_id(N'[tempdb].dbo.[#userroles\_kk]')

)

DROP TABLE #userroles\_kk

;

IF EXISTS

(

SELECT \* FROM tempdb.dbo.sysobjects

WHERE id = object\_id(N'[tempdb].dbo.[#rolemember\_kk]')

)

DROP TABLE #rolemember\_kk

;

IF EXISTS

(

SELECT \* FROM tempdb.dbo.sysobjects

WHERE id = object\_id(N'[tempdb].dbo.[##db\_name]')

)

DROP TABLE ##db\_name

;

DECLARE

@db\_name VARCHAR(255)

,@sql\_text VARCHAR(MAX)

;

SET @sql\_text =

'CREATE TABLE ##db\_name

(

LoginUserName VARCHAR(MAX)

,'

;

DECLARE cursDBs CURSOR FOR

SELECT [name]

FROM sys.databases

ORDER BY [name]

;

OPEN cursDBs

;

FETCH NEXT FROM cursDBs INTO @db\_name

WHILE @@FETCH\_STATUS = 0

BEGIN

SET @sql\_text =

@sql\_text + QUOTENAME(@db\_name) + ' VARCHAR(MAX)

,'

FETCH NEXT FROM cursDBs INTO @db\_name

END

CLOSE cursDBs

;

SET @sql\_text =

@sql\_text + 'IsSysAdminLogin CHAR(1)

,IsEmptyRow CHAR(1)

)'

--PRINT @sql\_text

EXEC (@sql\_text)

;

DEALLOCATE cursDBs

;

DECLARE

@RoleName VARCHAR(255)

,@UserName VARCHAR(255)

;

CREATE TABLE #permission

(

LoginUserName VARCHAR(255)

,databasename VARCHAR(255)

,[role] VARCHAR(255)

)

;

DECLARE cursSysSrvPrinName CURSOR FOR

SELECT [name]

FROM sys.server\_principals

WHERE

[type] IN ( 'S', 'U', 'G' )

AND principal\_id > 4

AND [name] NOT LIKE '##%'

AND [name] NOT IN

(

'transportuser'

,'ssmoltis'

,'rlobashov'

,'ReplUser'

,'OKTOGO\sqlrepl'

,'OKTOGO\sqlrepl'

)

ORDER BY [name]

;

OPEN cursSysSrvPrinName

;

FETCH NEXT FROM cursSysSrvPrinName INTO @UserName

WHILE @@FETCH\_STATUS = 0

BEGIN

CREATE TABLE #userroles\_kk

(

databasename VARCHAR(255)

,[role] VARCHAR(255)

)

;

CREATE TABLE #rolemember\_kk

(

dbrole VARCHAR(255)

,membername VARCHAR(255)

,membersid VARBINARY(2048)

)

;

DECLARE cursDatabases CURSOR FAST\_FORWARD LOCAL FOR

SELECT [name]

FROM sys.databases

ORDER BY [name]

;

OPEN cursDatabases

;

DECLARE

@DBN VARCHAR(255)

,@sqlText NVARCHAR(4000)

;

FETCH NEXT FROM cursDatabases INTO @DBN

WHILE @@FETCH\_STATUS = 0

BEGIN

SET @sqlText =

N'USE ' + QUOTENAME(@DBN) + ';

TRUNCATE TABLE #RoleMember\_kk

INSERT INTO #RoleMember\_kk

EXEC sp\_helprolemember

INSERT INTO #UserRoles\_kk

(DatabaseName,[Role])

SELECT db\_name(),dbRole

FROM #RoleMember\_kk

WHERE MemberName = ''' + @UserName + '''

'

--PRINT @sqlText ;

EXEC sp\_executesql @sqlText ;

FETCH NEXT FROM cursDatabases INTO @DBN

END

CLOSE cursDatabases

;

DEALLOCATE cursDatabases

;

INSERT INTO #permission

SELECT

@UserName 'user'

,b.name

,u.[role]

FROM

sys.sysdatabases b

LEFT JOIN

#userroles\_kk u

ON QUOTENAME(u.databasename) = QUOTENAME(b.name)

ORDER BY 1

;

DROP TABLE #userroles\_kk

;

DROP TABLE #rolemember\_kk

;

FETCH NEXT FROM cursSysSrvPrinName INTO @UserName

END

CLOSE cursSysSrvPrinName

;

DEALLOCATE cursSysSrvPrinName

;

TRUNCATE TABLE ##db\_name

;

DECLARE

@d1 VARCHAR(MAX)

,@d2 VARCHAR(MAX)

,@d3 VARCHAR(MAX)

,@ss VARCHAR(MAX)

;

DECLARE cursPermisTable CURSOR FOR

SELECT \* FROM #permission

ORDER BY 2 DESC

;

OPEN cursPermisTable

;

FETCH NEXT FROM cursPermisTable INTO @d1,@d2,@d3

WHILE @@FETCH\_STATUS = 0

BEGIN

IF NOT EXISTS

(

SELECT 1 FROM ##db\_name WHERE LoginUserName = @d1

)

BEGIN

SET @ss =

'INSERT INTO ##db\_name(LoginUserName) VALUES (''' + @d1 + ''')'

EXEC (@ss)

;

SET @ss =

'UPDATE ##db\_name SET ' + @d2 + ' = ''' + @d3 + ''' WHERE LoginUserName = ''' + @d1 + ''''

EXEC (@ss)

;

END

ELSE

BEGIN

DECLARE

@var NVARCHAR(MAX)

,@ParmDefinition NVARCHAR(MAX)

,@var1 NVARCHAR(MAX)

;

SET @var =

N'SELECT @var1 = ' + QUOTENAME(@d2) + ' FROM ##db\_name WHERE LoginUserName = ''' + @d1 + ''''

;

SET @ParmDefinition =

N'@var1 NVARCHAR(600) OUTPUT '

;

EXECUTE Sp\_executesql @var,@ParmDefinition,@var1 = @var1 OUTPUT

;

SET @var1 =

ISNULL(@var1, ' ')

;

SET @var =

' UPDATE ##db\_name SET ' + @d2 + '=''' + @var1 + ' ' + @d3 + ''' WHERE LoginUserName = ''' + @d1 + ''' '

;

EXEC (@var)

;

END

FETCH NEXT FROM cursPermisTable INTO @d1,@d2,@d3

END

CLOSE cursPermisTable

;

DEALLOCATE cursPermisTable

;

UPDATE ##db\_name SET

IsSysAdminLogin = 'Y'

FROM

##db\_name TT

INNER JOIN

dbo.syslogins SL

ON TT.LoginUserName = SL.[name]

WHERE

SL.sysadmin = 1

;

DECLARE cursDNamesAsColumns CURSOR FAST\_FORWARD LOCAL FOR

SELECT [name]

FROM tempdb.sys.columns

WHERE

OBJECT\_ID = OBJECT\_ID('tempdb..##db\_name')

AND [name] NOT IN ('LoginUserName','IsEmptyRow')

ORDER BY [name]

;

OPEN cursDNamesAsColumns

;

DECLARE

@ColN VARCHAR(255)

,@tSQLText NVARCHAR(4000)

;

FETCH NEXT FROM cursDNamesAsColumns INTO @ColN

WHILE @@FETCH\_STATUS = 0

BEGIN

SET @tSQLText =

N'UPDATE ##db\_name SET

IsEmptyRow = ''N''

WHERE IsEmptyRow IS NULL

AND ' + QUOTENAME(@ColN) + ' IS NOT NULL

;

'

--PRINT @tSQLText ;

EXEC sp\_executesql @tSQLText ;

FETCH NEXT FROM cursDNamesAsColumns INTO @ColN

END

CLOSE cursDNamesAsColumns

;

DEALLOCATE cursDNamesAsColumns

;

UPDATE ##db\_name SET

IsEmptyRow = 'Y'

WHERE IsEmptyRow IS NULL

;

UPDATE ##db\_name SET

IsSysAdminLogin = 'N'

FROM

##db\_name TT

INNER JOIN

dbo.syslogins SL

ON TT.LoginUserName = SL.[name]

WHERE

SL.sysadmin = 0

;

SELECT \* FROM ##db\_name

;

DROP TABLE ##db\_name

;

DROP TABLE #permission

;

END TRY

BEGIN CATCH

DECLARE

@cursDBs\_Status INT

,@cursSysSrvPrinName\_Status INT

,@cursDatabases\_Status INT

,@cursPermisTable\_Status INT

,@cursDNamesAsColumns\_Status INT

;

SELECT

@cursDBs\_Status = CURSOR\_STATUS('GLOBAL','cursDBs')

,@cursSysSrvPrinName\_Status = CURSOR\_STATUS('GLOBAL','cursSysSrvPrinName')

,@cursDatabases\_Status = CURSOR\_STATUS('GLOBAL','cursDatabases')

,@cursPermisTable\_Status = CURSOR\_STATUS('GLOBAL','cursPermisTable')

,@cursDNamesAsColumns\_Status = CURSOR\_STATUS('GLOBAL','cursPermisTable')

;

IF @cursDBs\_Status > -2

BEGIN

CLOSE cursDBs ;

DEALLOCATE cursDBs ;

END

IF @cursSysSrvPrinName\_Status > -2

BEGIN

CLOSE cursSysSrvPrinName ;

DEALLOCATE cursSysSrvPrinName ;

END

IF @cursDatabases\_Status > -2

BEGIN

CLOSE cursDatabases ;

DEALLOCATE cursDatabases ;

END

IF @cursPermisTable\_Status > -2

BEGIN

CLOSE cursPermisTable ;

DEALLOCATE cursPermisTable ;

END

IF @cursDNamesAsColumns\_Status > -2

BEGIN

CLOSE cursDNamesAsColumns ;

DEALLOCATE cursDNamesAsColumns ;

END

SELECT ErrorNum = ERROR\_NUMBER(),ErrorMsg = ERROR\_MESSAGE() ;

END CATCH

GO