CREATE TABLE [dbo].[JOBsExecutionCheckLog](

[ServerName] [varchar](255) NULL,

[JobName] [varchar](512) NULL,

[run\_date] [datetime] NULL,

[EmailSendTime] [datetime] NULL

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[JOBsExecutionTimeBenchmarks](

[ServerName] [varchar](255) NOT NULL,

[JobName] [varchar](512) NOT NULL,

[Approximate running time] [int] NOT NULL,

[min\_execution\_time] [int] NULL,

[max\_execution\_time] [int] NULL

) ON [PRIMARY]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

/\* Needed rights for linked server

USE [msdb]

GO

CREATE ROLE [check JOB]

GO

GRANT SELECT ON [dbo].[sysjobs] TO [check JOB]

GO

GRANT SELECT ON [dbo].[sysjobhistory] TO [check JOB]

GO

ALTER ROLE [check JOB] ADD MEMBER [Readonly]

GO

\*/

ALTER PROCEDURE [dbo].[CheckJobExecution]

( @recipients VARCHAR(MAX) = 'email',

@copy\_recipients VARCHAR(MAX) = '',

@profile\_name1 sysname = '' ,

@show\_all bit = 0,

@server\_list varchar(max) = 'servername'

)

AS

set nocount on

declare @html nvarchar(max),

@MailSubject NVARCHAR(255),

@Importance VARCHAR(6) = 'Normal'

declare @server\_list\_table table

(

ServerId int identity(1, 1)

, ServerName varchar(255)

)

insert into @server\_list\_table

select [Data] from [DbAdmin].[dbo].[SplitByDelim](@server\_list, ',')

where [Data] <> @@SERVERNAME

declare @job\_hystory table

(

[server] varchar(max) NOT NULL,

[job\_id] uniqueidentifier NOT NULL,

[job\_name] varchar(max) NOT NULL,

[run\_status] int NOT NULL,

[run\_date] int NOT NULL,

[run\_time] int NOT NULL,

[run\_duration] int NOT NULL,

[message] varchar(max) NOT NULL,

[operator\_emailed] int NULL,

[operator\_netsend] int NULL,

[operator\_paged] int NULL,

[retries\_attempted] int NOT NULL

)

declare @job\_hystory\_small table

(

[server] varchar(max) NOT NULL,

[job\_id] uniqueidentifier NOT NULL,

[job\_name] varchar(max) NOT NULL,

[run\_status] int NOT NULL,

[run\_date] datetime NOT NULL,

[seconds\_elapsed] float NOT NULL,

[message] varchar(max) NOT NULL

)

declare @job\_running table

(

[server] varchar(max) NOT NULL,

[job\_name] uniqueidentifier NOT NULL,

[running\_sec] int

)

insert into @job\_running

(

[server]

, [job\_name]

, [running\_sec]

)

SELECT

[server]

, [JobName]

, datediff(SECOND, StartDate, getdate()) as running\_sec

FROM

(

SELECT

@@SERVERNAME as [server]

, JobName = sj.job\_id

, StartDate = sja.start\_execution\_date

, EndDate = sja.stop\_execution\_date

, Status = CASE

WHEN ISNULL(sjh.run\_status,-1) = -1 AND sja.start\_execution\_date IS NULL AND sja.stop\_execution\_date IS NULL THEN 'Idle'

WHEN ISNULL(sjh.run\_status,-1) = -1 AND sja.start\_execution\_date IS NOT NULL AND sja.stop\_execution\_date IS NULL THEN 'Running'

WHEN ISNULL(sjh.run\_status,-1) =0 THEN 'Failed'

WHEN ISNULL(sjh.run\_status,-1) =1 THEN 'Succeeded'

WHEN ISNULL(sjh.run\_status,-1) =2 THEN 'Retry'

WHEN ISNULL(sjh.run\_status,-1) =3 THEN 'Canceled'

END

FROM MSDB.DBO.sysjobs sj

JOIN MSDB.DBO.sysjobactivity sja

ON sj.job\_id = sja.job\_id

JOIN (SELECT MaxSessionid = MAX(Session\_id) FROM MSDB.DBO.syssessions) ss

ON ss.MaxSessionid = sja.session\_id

LEFT JOIN MSDB.DBO.sysjobhistory sjh

ON sjh.instance\_id = sja.job\_history\_id

) running\_jobs

where [Status] = 'Running'

declare @job\_exec\_time\_average table

(

[server] varchar(max) NOT NULL,

[job\_id] uniqueidentifier NOT NULL,

[seconds\_elapsed\_average] float NOT NULL

)

declare @job\_error\_list table

(

[server] varchar(max) NOT NULL,

[job\_id] uniqueidentifier NOT NULL,

[job\_name] varchar(max) NOT NULL,

[run\_status] int NOT NULL,

[run\_date] datetime NOT NULL,

[seconds\_elapsed] float NOT NULL,

[message] varchar(max) NOT NULL,

[seconds\_elapsed\_average] float NOT NULL,

[error\_description] varchar(max) NOT NULL

)

-- load hystory from current server

INSERT INTO @job\_hystory

(

[server], [job\_id], [job\_name], [run\_status], [run\_date], [run\_time],

[run\_duration], [message], [operator\_emailed], [operator\_netsend], [operator\_paged], [retries\_attempted]

)

--exec msdb..sp\_help\_jobhistory

SELECT

[server],

sjh.[job\_id],

sj.[name] AS [job\_name],

[run\_status],

[run\_date],

[run\_time],

[run\_duration],

[message],

[operator\_id\_emailed],

[operator\_id\_netsent],

[operator\_id\_paged],

[retries\_attempted]

FROM [msdb].[dbo].[sysjobhistory] sjh

LEFT JOIN [msdb].[dbo].[sysjobs] sj

ON sjh.[job\_id] = sj.[job\_id]

WHERE [step\_id] = 0

-- load hystory from linked servers

declare @cmd varchar(max)

declare @sid int = 1

while @sid<= (select max(ServerId) from @server\_list\_table)

begin

set @cmd = 'SELECT

[server]

, sjh.[job\_id]

, sj.[name] AS [job\_name]

, [run\_status]

, [run\_date]

, [run\_time]

, [run\_duration]

, [message]

, [operator\_id\_emailed]

, [operator\_id\_netsent]

, [operator\_id\_paged]

, [retries\_attempted]

FROM '+(select ServerName from @server\_list\_table where ServerId = @sid)+'.[msdb].[dbo].[sysjobhistory] sjh

LEFT JOIN '+(select ServerName from @server\_list\_table where ServerId = @sid)+'.[msdb].[dbo].[sysjobs] sj

ON sjh.[job\_id] = sj.[job\_id]

WHERE [step\_id] = 0'

insert into @job\_hystory

exec(@cmd)

set @cmd =

'SELECT

[server]

, [JobName]

, datediff(SECOND, StartDate, getdate()) as running\_sec

FROM

(

SELECT

@@SERVERNAME as [server]

, JobName = sj.job\_id

, StartDate = sja.start\_execution\_date

, EndDate = sja.stop\_execution\_date

, Status = CASE

WHEN ISNULL(sjh.run\_status,-1) = -1 AND sja.start\_execution\_date IS NULL AND sja.stop\_execution\_date IS NULL THEN ''Idle''

WHEN ISNULL(sjh.run\_status,-1) = -1 AND sja.start\_execution\_date IS NOT NULL AND sja.stop\_execution\_date IS NULL THEN ''Running''

WHEN ISNULL(sjh.run\_status,-1) =0 THEN ''Failed''

WHEN ISNULL(sjh.run\_status,-1) =1 THEN ''Succeeded''

WHEN ISNULL(sjh.run\_status,-1) =2 THEN ''Retry''

WHEN ISNULL(sjh.run\_status,-1) =3 THEN ''Canceled''

END

FROM MSDB.DBO.sysjobs sj

JOIN MSDB.DBO.sysjobactivity sja

ON sj.job\_id = sja.job\_id

JOIN (SELECT MaxSessionid = MAX(Session\_id) FROM MSDB.DBO.syssessions) ss

ON ss.MaxSessionid = sja.session\_id

LEFT JOIN MSDB.DBO.sysjobhistory sjh

ON sjh.instance\_id = sja.job\_history\_id

) running\_jobs

where [Status] = ''Running'''

insert into @job\_running

(

[server]

, [job\_name]

, [running\_sec]

)

exec(@cmd)

set @sid = @sid+1

end

-- aggregate information

INSERT INTO @job\_hystory\_small

(

[server], [job\_id], [job\_name], [run\_status], [run\_date], [seconds\_elapsed], [message]

)

SELECT

[server],

[job\_id],

[job\_name],

[run\_status],

DATEADD(SS,

(([run\_time] % 1000000) / 10000) \* 3600 +

(([run\_time] % 10000) / 100) \* 60 +

([run\_time] % 100)

,CONVERT(datetime,(CONVERT(VARCHAR,[run\_date])),104)) AS [run\_date],

(([run\_duration] % 1000000) / 10000) \* 3600 +

(([run\_duration] % 10000) / 100) \* 60 +

([run\_duration] % 100) AS [SECONDS\_ELAPSED]

,[message]

FROM @job\_hystory

where job\_name <> 'CkeckJobActivity'

GROUP BY

[job\_id],

[job\_name],

[run\_status],

[run\_date],

[run\_time],

[run\_duration],

[server],

[message]

-- get average info about working-time of JOB`s.

INSERT INTO @job\_exec\_time\_average

SELECT

[server],

[job\_id],

SUM(CASE [run\_status]

WHEN 1 THEN 1

ELSE 0

END \* [seconds\_elapsed])

/

CASE WHEN SUM(CASE [run\_status]

WHEN 1 THEN 1

ELSE 0

END) = 0 THEN 0.0000001

ELSE SUM(CASE [run\_status]

WHEN 1 THEN 1

ELSE 0

END)

END AS [seconds\_elapsed\_average\_weighted]

FROM @job\_hystory\_small

WHERE

DATEDIFF(dd,[run\_date],GETDATE()) >= 1

GROUP BY

[server],

[job\_id]

-- Find Errors and exceptions

INSERT INTO @job\_error\_list

SELECT

jh.\*,

ISNULL(jt.[seconds\_elapsed\_average],0),

CASE

WHEN jh.[run\_status] = 0 THEN 'There are errors in JOB.'

WHEN jh.[run\_status] = 2 THEN 'One of steps was reruning'

WHEN jh.[run\_status] = 3 THEN 'Job was cancelled'

WHEN jh.[run\_status] = 5 THEN 'Unknown status'

WHEN jh.[run\_status] = 1 AND

jh.[seconds\_elapsed] /

CASE

WHEN ISNULL(jt.[seconds\_elapsed\_average],0) = 0 THEN 0.0000001

ELSE jt.[seconds\_elapsed\_average]

END >= 1.2 and jh.[seconds\_elapsed]<> 0 and ISNULL(jt.[seconds\_elapsed\_average],0) <> 0 and (select isnull(min\_execution\_time, max\_execution\_time) from [DbAdmin].[dbo].[JOBsExecutionTimeBenchmarks] JE with (nolock) where JE.ServerName = jh.[server] and JE.JobName = jh.job\_name) is null

THEN 'Job was working more time then offen'

WHEN jh.[run\_status] = 1 AND

jh.[seconds\_elapsed] /

CASE

WHEN ISNULL(jt.[seconds\_elapsed\_average],0) = 0 THEN 0.0000001

ELSE jt.[seconds\_elapsed\_average]

END <= 0.8 and jh.[seconds\_elapsed]<> 0 and ISNULL(jt.[seconds\_elapsed\_average],0) <> 0 and (select isnull(min\_execution\_time, max\_execution\_time) from [DbAdmin].[dbo].[JOBsExecutionTimeBenchmarks] JE with (nolock) where JE.ServerName = jh.[server] and JE.JobName = jh.job\_name) is null

THEN 'Job was working less time then offen'

WHEN jh.[run\_status] = 1 AND

jh.[seconds\_elapsed] not between (select min\_execution\_time from [DbAdmin].[dbo].[JOBsExecutionTimeBenchmarks] JE with (nolock) where JE.ServerName = jh.[server] and JE.JobName = jh.job\_name) and (select max\_execution\_time from [DbAdmin].[dbo].[JOBsExecutionTimeBenchmarks] JE with (nolock) where JE.ServerName = jh.[server] and JE.JobName = jh.job\_name)

THEN 'greater then Approximate'

ELSE 'Succesfull'

END AS [error\_description]

FROM @job\_hystory\_small jh

LEFT JOIN @job\_exec\_time\_average jt

ON jh.[server] = jt.[server] AND

jh.[job\_id] = jt.[job\_id]

WHERE

jh.[run\_date] = (SELECT MAX([run\_date]) FROM @job\_hystory\_small t

WHERE

t.[server] = jh.[server] AND

t.[job\_id] = jh.[job\_id] )

insert into @job\_error\_list

(

[server],

[job\_name],

[error\_description]

)

select

jh.[server]

, jh.job\_name

, 'job is running now, but it very long'

from @job\_running jh

LEFT JOIN @job\_exec\_time\_average jt

ON jh.[server] = jt.[server] AND

jh.job\_name = jt.[job\_id]

where jh.running\_sec - jt.seconds\_elapsed\_average < 0

UPDATE @job\_error\_list

SET [run\_status] = 2, [error\_description] = 'One of steps was reruning'

FROM @job\_error\_list el

WHERE EXISTS(SELECT \* FROM [msdb].[dbo].[sysjobhistory] sjh

WHERE

sjh.[server] = el.[server] AND

sjh.[job\_id] = el.[job\_id] AND

(sjh.[run\_date] = YEAR(el.[run\_date]) \* 10000 + MONTH(el.[run\_date]) \* 100 + DAY(el.[run\_date])) AND

sjh.[run\_status] = 2)

select distinct [server], [job\_id]

into #t

from @job\_error\_list

WHERE run\_status in (0, 2)

select

[run\_date]

, [message]

, [server]

, [job\_id]

, [step\_id]

, [run\_status]

into #errors

from [server].[msdb].[dbo].[sysjobhistory]

where [job\_id] in (select [job\_id] from #t where [server] = @@servername )

set @sid = 1

while @sid<= (select max(ServerId) from @server\_list\_table)

begin

set @cmd = 'select

[run\_date]

, [message]

, [server]

, [job\_id]

, [step\_id]

, [run\_status]

from '+(select ServerName from @server\_list\_table where ServerId = @sid)+'.[msdb].[dbo].[sysjobhistory]

where [job\_id] in (select [job\_id] from #t where [server] = '''+(select ServerName from @server\_list\_table where ServerId = @sid)+''' )'

insert into #errors

exec (@cmd)

set @sid = @sid+1

end

SET @MailSubject = N'Status of JOB`s working time.'-- + case when (SELECT COUNT(\*) FROM @job\_error\_list WHERE [run\_status] = 0) > 0 then ' Critical errors was finded!' else ' Trere are no critical' end

--select \* from @job\_error\_list return

IF (SELECT COUNT(\*) FROM @job\_error\_list) > 0

BEGIN

if @show\_all = 0

begin

delete from @job\_error\_list where [run\_status] = 1 AND [error\_description] like '%succesful%'

end

SET @html =

N'<table border="0" style="margin-left: auto; margin-right: auto; margin-top: 0px; max-width: 800px; padding: 70px 30px 50px 30px; background-color: white;"><tr><td>' +

N'<table border="0"><tr><td><img src="http://servicedesk:8080/custom/customimages/Custom\_HeadLogo.gif?1493304584540" width="200" height="58" style="margin: 7px 7px 7px 0;"></td><td style="padding: 10px 0px 0px 20px;"><h1><font face="Cambria" color="#383484" size="4">JOB`s working time and problems</font></h1></td></tr></table>' +

N'<h2><font face="Calibri" color="#000000" size="3" style="font-weight: normal;">During the scheduled check of the correctness of the SQL Server Agent service execution, the following deviations were detected:</font></h2>' +

N'<table border="1" bordercolor="#a6a6a6" cellpadding="0" cellspacing="0" style="font:12pt sans-serif; font-family: Calibri, Arial;">' +

N'<tr>' +

N'<th bgcolor="#f1f1f1" style="padding: 5px; font-weight: normal;">Server name</th>' +

N'<th bgcolor="#f1f1f1" style="padding: 5px; font-weight: normal;">JOB name</th>' +

--N'<th bgcolor="#f1f1f1" style="padding: 5px; font-weight: normal;">Status</th>' +

N'<th bgcolor="#f1f1f1" style="padding: 5px; font-weight: normal;">Begin time</th>' +

N'<th bgcolor="#f1f1f1" style="padding: 5px; font-weight: normal;">Time elapsed, sec</th>' +

N'<th bgcolor="#f1f1f1" style="padding: 5px; font-weight: normal;">TimeBenchmark, sec</th>' +

N'<th bgcolor="#f1f1f1" style="padding: 5px; font-weight: normal;">Average time by 10 last runs, sec</th>' +

N'<th bgcolor="#f1f1f1" style="padding: 5px; font-weight: normal;">Pass Approximate (Y/N)</th>' +

N'<th bgcolor="#f1f1f1" style="padding: 5px; font-weight: normal;">Result</th>' +

N'</tr>' +

isnull(replace(replace(CAST(

(SELECT

N'<td style="padding: 5px;">' + ISNULL([server], N'') + N'</td>' +

N'<td style="padding: 5px;">' + ISNULL([job\_name], N'') + N'</td>' +

--N'<td style="padding: 5px;">' + CASE

-- WHEN [run\_status] = 0 THEN N'With errors'

-- WHEN [run\_status] = 2 THEN N'One of step was reruned'

-- WHEN [run\_status] = 3 THEN N'Cancelled'

-- WHEN [run\_status] = 5 THEN N'Unknown'

-- ELSE N'With out errors'

--END + N'</td>' +

N'<td style="padding: 5px;">' + ISNULL(CONVERT(nvarchar,[run\_date],120), N'') + N'</td>' +

N'<td style="padding: 5px; text-align: center;">' + ISNULL(CONVERT(nvarchar,round([seconds\_elapsed],2)), N'') + N'</td>' +

N'<td style="padding: 5px; text-align: center;">' + concat((select CONVERT(nvarchar,[min\_execution\_time]) from [DbAdmin].[dbo].[JOBsExecutionTimeBenchmarks] JE with (nolock) where JE.ServerName = jh.[server] and JE.JobName = jh.job\_name), ' - ', (select CONVERT(nvarchar,[max\_execution\_time]) from [DbAdmin].[dbo].[JOBsExecutionTimeBenchmarks] JE with (nolock) where JE.ServerName = jh.[server] and JE.JobName = jh.job\_name)) + N'</td>' +

N'<td style="padding: 5px; text-align: center;">' + ISNULL(CONVERT(nvarchar,round([seconds\_elapsed\_average],2)), N'') + N'</td>' +

N'<td style="padding: 5px; text-align: center;">' + case

when [run\_status] = 1 AND [error\_description] like '%Approximate%' then N'NO'

when [run\_status] = 1 AND [error\_description] not like '%Approximate%' and concat((select CONVERT(nvarchar,[min\_execution\_time]) from [DbAdmin].[dbo].[JOBsExecutionTimeBenchmarks] JE with (nolock) where JE.ServerName = jh.[server] and JE.JobName = jh.job\_name), ' - ', (select CONVERT(nvarchar,[max\_execution\_time]) from [DbAdmin].[dbo].[JOBsExecutionTimeBenchmarks] JE with (nolock) where JE.ServerName = jh.[server] and JE.JobName = jh.job\_name)) = N' - ' THEN N'Unknown'

when [run\_status] <> 1 then N'errors'

ELSE N'YES' END + N'</td>' +

CASE

WHEN [run\_status] = 0 THEN N'<td style="padding: 5px; color: #ac0c0c;" bgcolor="#fb8484"><b>' + [error\_description] + N'</b></td>'

WHEN [run\_status] = 2 THEN N'<td style="padding: 5px;" bgcolor="#f2f09d">' + [error\_description] + N'</td>'

WHEN [run\_status] = 3 THEN N'<td style="padding: 5px; color: #ac0c0c;" bgcolor="#fb8484"><b>' + [error\_description] + N'</b></td>'

WHEN [run\_status] = 5 THEN N'<td style="padding: 5px; color: #ac0c0c;" bgcolor="#fb8484"><b>' + [error\_description] + N'</b></td>'

WHEN [run\_status] = 1 AND [error\_description] like '%succesful%' THEN N'<td style="padding: 5px; color: #000000;" bgcolor="#98FB98"><b>' + [error\_description] + N'</b></td>'

WHEN [run\_status] = 1 AND [error\_description] like '%less%' THEN N'<td style="padding: 5px;" bgcolor="#f2f09d">' + [error\_description] + N'</td>'

WHEN [run\_status] = 1 AND [error\_description] like '%error%' THEN N'<td style="padding: 5px;" bgcolor="#b7f29d">' + [error\_description] + N'</td>'

WHEN [run\_status] = 1 AND [error\_description] like '%Approximate%' THEN N'<td style="padding: 5px;" bgcolor="#B0E0E6">' + [error\_description] + N'</td>'

WHEN [run\_status] = 1 AND [error\_description] like '%more%' THEN N'<td style="padding: 5px; color: #000000;" bgcolor="#FFDAB9"><b>' + [error\_description] + N'</b></td>'

END

FROM @job\_error\_list jh

WHERE [run\_status] <> 4

and not exists (

select top 1 1 from [DbAdmin].[dbo].[JOBsExecutionCheckLog] t (nolock) where t.[ServerName] = jh.[server] and t.[JobName] = jh.job\_name and t.[run\_date] = jh.[run\_date]

)

FOR XML PATH('tr'), TYPE)

AS NVARCHAR(MAX)), '&gt;', '>'), '&lt;', '<'), N'') +

N'</table>'

if exists( select top 1 1

FROM @job\_error\_list el

LEFT JOIN #errors sjh ON

el.[server] = sjh.[server] AND

el.[job\_id] = sjh.[job\_id] AND

YEAR(el.[run\_date]) \* 10000 + MONTH(el.[run\_date]) \* 100 + DAY(el.[run\_date]) = sjh.[run\_date]

WHERE

el.[run\_status] = 0 AND

sjh.[run\_status] = 0 AND

sjh.[step\_id] > 0

and not exists (

select top 1 1 from [DbAdmin].[dbo].[JOBsExecutionCheckLog] t (nolock) where t.[ServerName] = el.[server] and t.[JobName] = el.job\_name and t.[run\_date] = el.[run\_date]

)

)

begin

IF (SELECT COUNT(\*) FROM @job\_error\_list WHERE [run\_status] = 0) > 0

BEGIN

SET @Importance = 'High'

SET @html +=

N'<h2><font face="Calibri" color="#000000" size="3" style="font-weight: normal;">Text of critical errors:</font></h2>' +

N'<table border="1" bordercolor="#a6a6a6" cellpadding="0" cellspacing="0" style="font:12pt sans-serif; max-width: 800px; font-family: Calibri, Arial;">' +

N'<tr>' +

N'<th bgcolor="#f1f1f1" style="padding: 5px; font-weight: normal;">Server Name</th>' +

N'<th bgcolor="#f1f1f1" style="padding: 5px; font-weight: normal;">JOB name</th>' +

N'<th bgcolor="#f1f1f1" style="padding: 5px; font-weight: normal;">Error text</th>' +

N'<th bgcolor="#f1f1f1" style="padding: 5px; font-weight: normal;">Error text from last step</th>' +

N'</tr>' +

isnull(replace(replace(CAST(

(SELECT

N'<td style="padding: 5px;">' + ISNULL(el.[server], N'') + N'</td>' +

N'<td style="padding: 5px;">' + ISNULL(el.[job\_name], N'') + N'</td>' +

N'<td style="padding: 5px;">' + ISNULL(el.[message], N'') + N'</td>' +

N'<td style="padding: 5px;">' + ISNULL(sjh.[message], N'') + N'</td>'

FROM @job\_error\_list el

LEFT JOIN #errors sjh ON

el.[server] = sjh.[server] AND

el.[job\_id] = sjh.[job\_id] AND

YEAR(el.[run\_date]) \* 10000 + MONTH(el.[run\_date]) \* 100 + DAY(el.[run\_date]) = sjh.[run\_date]

WHERE

el.[run\_status] = 0 AND

sjh.[run\_status] = 0 AND

sjh.[step\_id] > 0

and not exists (

select top 1 1 from [DbAdmin].[dbo].[JOBsExecutionCheckLog] t (nolock) where t.[ServerName] = el.[server] and t.[JobName] = el.job\_name and t.[run\_date] = el.[run\_date]

)

FOR XML PATH('tr'), TYPE)

AS NVARCHAR(MAX)), '&gt;', '>'), '&lt;', '<'), N'') +

N'</table>'

END

end

SET @html += N'<hr color="#e9e9e9" style="margin-top: 40px;"/>' +

N'<table border="0"><tr><td><font face="Calibri" color="#000000" size="3" style="font-weight: normal;">Contact email: InfrastructureDBA@travelrepublic.co.uk</font></td></tr>' +

N'<tr><td><font face="Calibri" color="#000000" size="3" style="font-weight: normal;">Telephone: +7 (911) 144-56-65 (Vladimir)</font></td></tr></table>' +

N'</td></tr></table>'

END

ELSE BEGIN

RETURN

END

DECLARE @MyTableVar TABLE

(

[ServerName] varchar(255)

, [JobName] varchar(512)

, [run\_date] datetime

, [EmailSendTime] datetime

);

insert into [DbAdmin].[dbo].[JOBsExecutionCheckLog]

(

[ServerName]

, [JobName]

, [run\_date]

, [EmailSendTime]

)

output inserted.\* into @MyTableVar

select

jh.[server]

, jh.job\_name

, jh.run\_date

, getdate()

FROM @job\_error\_list jh

WHERE [run\_status] <> 4

and not exists (

select top 1 1 from [DbAdmin].[dbo].[JOBsExecutionCheckLog] t (nolock) where t.[ServerName] = jh.[server] and t.[JobName] = jh.job\_name and t.[run\_date] = jh.[run\_date]

)

IF @profile\_name1 = '' SET @profile\_name1 = (SELECT TOP 1 name FROM msdb.dbo.sysmail\_profile)

if exists (

select top 1 1 from @MyTableVar

)

begin

EXEC msdb..sp\_send\_dbmail

@recipients = @recipients,

@copy\_recipients = @copy\_recipients,

@subject = @MailSubject,

@body = @html,

@body\_format = 'HTML',

@profile\_name = @profile\_name1,

@importance = @Importance

print 'mail was send!'

end