**Alertas**

**Processo Bloqueado**

**Arquivo de Log FULL**

**Espaço em Disco**

**Tempdb Utilização Arquivo**

**Conexão SQL Server**

**Status Database**

**Banco de Dados Corrompido**

**Página Corrompida**

**Queries Demoradas**

**Falha de Job**

**SQL Server Reiniciado**

**Database Criada**

**Database sem Backup**

**Alertas de Severidade**

1. **Pré-Requisitos**

Criar a procedure sp\_whoisactive (Adam Machanic)

* 1. **Link to Download:**

http://whoisactive.com/downloads/

* 1. **Testar procedure Whoisactive**

exec sp\_whoisactive

**Obs**. Antes de continuar, verifique se o envio de e-mail na sua instância SQL Server está funcionando.

**DatabaseMail**

1. Criar um **operador** para utilização dos **jobs** e uma **database** para armazenar os dados

USE [msdb]

if not exists (

select NULL

from msdb.dbo.sysoperators

where name = 'DTX\_DBA\_Operator')

begin

EXEC [msdb].[dbo].[sp\_add\_operator]

@name = N'DTX\_DBA\_Operator',

@enabled = 1,

@pager\_days = 0,

@email\_address = N'EMail1@provedor.com' -- To put more Emails: ‘suporte@dataex.com; EMail2@provedor.com' (Email 2 pode ser dos Clientes)

end

-- Somente para o Time DBA

USE [msdb]

if not exists (

select NULL

from msdb.dbo.sysoperators

where name = 'DTX\_DBA\_Team\_Operator')

begin

EXEC [msdb].[dbo].[sp\_add\_operator]

@name = DTX\_DBA\_Team\_Operator',

@enabled = 1,

@pager\_days = 0,

@email\_address = N'DBATEAM@provedor.com' -- ‘EMail1@provedor.com;EMail2@provedor.com'

end

go

Criar uma database para armazenar as informações.

CREATE DATABASE [DBADataex]

ON PRIMARY (

NAME = N'DBADataex', FILENAME = N'C:\...\DBADataex.mdf' , SIZE = 102400KB , FILEGROWTH = 102400KB

)

LOG ON (

NAME = N'DBADataex\_log', FILENAME = N'C:\...\DBADataex\_log.ldf' SIZE = 30720KB , FILEGROWTH = 30720KB

)

GO

ALTER DATABASE [DBADataex] SET RECOVERY SIMPLE

Tabela para ignorar algumas bases não importantes de algumas rotinas no ambiente, como por exemplo o **Checkdb**.

USE DBADataex

IF ( OBJECT\_ID('[dbo].[dtx\_tb\_Ignore\_Databases]') IS NOT NULL )

DROP TABLE [dbo].dtx\_tb\_Ignore\_Databases

CREATE TABLE [dbo].[dtx\_tb\_Ignore\_Databases] (

[Nm\_Database] VARCHAR(500)

)

**Se já tiver alguma base para ser ignorada, insira aqui:**

INSERT INTO [dtx\_tb\_Ignore\_Databases]

VALUES('Nm\_Database1'),('Nm\_Database2'), ('Nm\_Database3')

1. **Executer Script**: **Create Alert Table.sql**

Execute a procedure para criar a tabela com todos as configurações de alertas.

Mude o parâmetro "suporte@provedor.com;metodo@provedor.com" para os e-mails que vão receber os alertas.

Também mude o parâmetro @Profile para o Database Mail Profile configurado no seu SQL Server.

USE DBADataex

exec dtx\_sp\_Configuration\_Table 'Email1@provedor.com;Email2@provedor.com', @Profile, @Fl\_Language (1 - Portuguese | 0 -- English)

**Check the Parameters**

select \* from [dbo].Alert\_Parameter

1. **Executer Script:** **Create All Alert Procedures and Jobs.sql**

Criar os alertas de severidade.

EXEC dtx\_sp\_Alert\_Severity

-- Utilize esses scripts para testar os alertas já criados

-- Use this script to test some of the created alerts

EXEC dbo.stpAlert\_Every\_Day

EXEC dbo.stpTest\_Alerts

--------------------------------------------------------------------------------------------------------------------------------

-- 3) Se tiver interesse, pode criar algumas rotinas adicionais para o seu banco de dados. Você tem que entender o que essas rotinas fazem antes de criar. Tenha cuidado. Se não conhece SQL Server talvez seja melhor pular esse item 3.0 e ir para o item 4 dos scripts.

-- 3) If you wish, you can create additional jobs and Alerts for your databases. But be carefull, you must be aware about what these routines do before you create it. Also be carefull, If you are not a SQL Server DBA or if you don't have a good doaming of it, maybe you should skip to step 4 of this setup.

--------------------------------------------------------------------------------------------------------------------------------

\*\*\*\*\*\* PT BR-> Na dúvida acesse os vídeos explicando cada uma dessas rotinas \*\*\*\*\*\*

link:

"\*\*\* If you are not confident enough or if it's still not clear, watch below videos about each of these routines.\*\*\*"

link: English video not available yet

---------------------3.1) Job to execute a checkdb on databases and an alert if we have some corrupted database.

--Script: "3.1 - CheckDB - Job and Alert.sql"

---------------------3.2) Profile to monitor what is taking more than 3 seconds to run and an alert if that number is too high in the last five minutes

--Script: "3.2 - Profile Duration - Job and Alert.sql"

-- obs.: Feel free to change for a XEvents here

select \* FROM fn\_trace\_getinfo (null)

------ Test the server side traces

waitfor delay '00:00:05'

--Execute the job

EXEC msdb.dbo.sp\_start\_job N'DBA - Load Server Side Trace';

--Confira o resultado

select \* from Traces..Queries\_Profile

---------------------3.3) XEvent to monitor database errors and a daily alert information about them

--Script: "3.3 - XEvent Error - Job and Alert.sql"

select 1/0

--Run the job

EXEC msdb.dbo.sp\_start\_job N'DBA - Load XEvent Database Error';

select \* from Traces..Log\_DB\_Error

---------------------3.4) Store information about index fragmentation daily

--Script: "3.4 - Index Fragmentation History.sql"

--Open the procedure to execute and test (remove the 6 am lock to test)

SELECT \* FROM [dbo].[Index\_Fragmentation\_History]

---------------------3.5) XEvent to monitor database Dealocks and a daily alert information about them

--Script: "3.5 - Deadlock - Job and Alert.sql"

--To Test

create table test1 (id int)

insert into test1 values (1)

create table test2 (id int)

insert into test2 values (2)

-- Connection 1

BEGIN TRAN

UPDATE test1

SET id = id

UPDATE test2

SET id = id

--commit

-- Connection 2

BEGIN TRAN

UPDATE test2

SET id = id

UPDATE test1

SET id = id

EXEC msdb.dbo.sp\_start\_job N'DBA - Load XEvent Deadlock'

SELECT \* FROM Traces.[dbo].[Log\_DeadLock]

DROP TABLE teste1

DROP TABLE teste2

---------------------3.6) Log to monitor the whoisactive every minute

--Script: "3.6 - Log Whoisactive.sql"

--Test in another connection

WAITFOR DELAY '00:01:00'

--run the job

EXEC msdb.dbo.sp\_start\_job N'DBA - Load Whoisactive'

SELECT \* FROM dbo.Log\_Whoisactive

---------------------3.7) If you have a Database Mirroring

--Script: "3.7 - Alert Database Mirroring.sql"

---------------------3.8) If you have a AlwaysON AG

--Script: "3.8 - Alert AlwaysON AG.sql"

---------------------3.9) If you have a Failover Cluster (do not work for SQL 2008 or less)

--Script: "3.9 - Alert Failover Cluster.sql"

--------------------------------------------------------------------------------------------------------------------------------

-- 4) Execute este script para criar o checklist do banco de dados

-- 4) Execute this script to create the database checklist

--------------------------------------------------------------------------------------------------------------------------------

-- RUN Script: "4.0 - Procedures CheckList.sql"

-- Run the checklist job to test

EXEC msdb.dbo.sp\_start\_job N'DBA - CheckList SQL Server Instance';

-- Finish!!!

---------------------------------------------------------------------------------------------------------------------------------

-- Scripts Just for Managed Instance

---------------------------------------------------------------------------------------------------------------------------------

-- If you are creating this scripts on an Azure Managed Instance, you need to disable some alerts and checklist informations.

-- Disable Alerts

update Alert\_Parameter

set Fl\_Enable = 0

WHERE Nm\_Alert IN ('SQL Server Configuration','Database Without Backup','CPU Utilization','Disk Space',

'Slow File Growth','Database Without Log Backup','Memory Available')

--Disable CheckList Information

UPDATE CheckList\_Parameter

SET Fl\_Enabled = 0

WHERE Nm\_Procedure IN ('stpCheckList\_Disk\_Space' ,'stpCheckList\_AutoGrowth','stpCheckList\_Database\_Without\_Backup','stpCheckList\_Backup\_Executed','stpCheckList\_Traces\_Queries')

-- Enable

update Alert\_Parameter

set Fl\_Enable = 1

WHERE Nm\_Alert IN ('CPU Utilization MI')