**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'suporte@dataex.com.br' -- 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'suporte@dataex.com.br' -- ‘EMail1@provedor.com;EMail2@provedor.com'

end

go

Criar uma database para armazenar as informações.

CREATE DATABASE [ ]

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@dataex.com.br" 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 'suporte@dataex.com.br', @Profile, @Fl\_Language -- (1 - Portuguese | 0 -- English)

**Check the Parameters**

select \* from [dbo].dtx\_tb\_Alert\_Parameter

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

Criar os alertas de severidade.

EXEC dtx\_sp\_Alert\_Severity

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

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

--Execute the job

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

--Confira o resultado

**select \* from dtx\_tb\_Queries\_Profile**

**Executer Script: 3.3 - XEvent Error - Job and Alert.sql**

Teste

select 1/0

select \* from dtx\_tb\_Log\_DB\_Error

**Executer Script: 3.4 - Index Fragmentation History.sql**

**Executer Script: 3.5 - Deadlock - Job and Alert.sql**

Testes

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'DataEX - Load XEvent Deadlock'**

**SELECT \* FROM [dbo].[dtx\_tb\_Log\_DeadLock]**

**Executer Script: 3.6 - Log Whoisactive.sql**

SELECT \* FROM dbo.dtx\_tb\_Log\_Whoisactive