

Configure e Reconfigure

SP_CONFIGURE não aceita novos valores de opção de configuração fora dos intervalos válidos documentados para cada opção de configuração.

Exibe ou altera parâmetros de configuração global para o servidor atual.

O padrão das permissões de CONFIGURE e RECONFIGURE são concedidos aos possuidores da permissão ALTER SETTINGS. As funções de servidor fixas **sysadmin** e **serveradmin** contêm esta permissão implicitamente.

Atualiza o valor configurado atualmente de uma opção de configuração alterada com o procedimento armazenado do sistema SP_CONFIGURE. Como algumas opções de configuração requerem uma parada e um reinício do servidor para atualizar o valor em execução no momento.

```
EXEC SP_CONFIGURE;

USE MASTER
EXEC sp_configure 'show advanced option', '1';

RECONFIGURE;
EXEC SP_CONFIGURE;

SELECT a.name, a.value, a.value_in_use
from sys.configurations a;

SELECT a.name, a.value, a.value_in_use
from sys.configurations a
where a.name = 'max server memory (MB)';

exec sp_configure 'show advanced options','1';
reconfigure;
exec sp_configure 'max server memory', 2147483647; --2147483647
reconfigure;
exec sp_configure 'show advanced options', 0;
reconfigure;
```

Stored Procedures no SQL Server

Stored Procedure, que traduzido significa Procedimento Armazenado, é uma conjunto de comandos em **SQL** que podem ser executados de uma só vez, como em uma função. Ele armazena tarefas repetitivas e aceita parâmetros de entrada para que a tarefa seja efetuada de acordo com a necessidade individual.

Um Stored Procedure pode reduzir o tráfego na rede, melhorar a performance de um banco de dados, criar tarefas agendadas, diminuir riscos, criar rotinas de processamento, etc.

Por todas estas e outras funcionalidades é que os stored procedures são de extrema importância para os DBAs e desenvolvedores.

Quando utilizar procedures

- Quando temos várias aplicações escritas em diferentes linguagens, ou rodam em plataformas diferentes, porém executam a mesma função.
- Quando damos prioridade à consistência e segurança.

Os bancos (Itaú, Bradesco, Real, etc), por exemplo, em geral, utilizam stored procedures para todas as operações em comum. Os procedimentos podem assegurar que as operações sejam registradas de forma correta e segura.

Por que é mais seguro?

Seguindo a linha de raciocínio dos bancos, **utilizando stored procedures outras aplicações e usuários não conseguiriam nenhum tipo de acesso às tabelas do banco de dados** de forma direta.

Eles poderiam apenas executar os stored procedures, que rodam ações específicas e determinadas pelos DBAs e desenvolvedores.

Criando um stored procedure

Este é um exemplo de um stored procedure que executa uma consulta utilizando um filtro por descrição, em uma tabela específica de nosso banco de dados.

```
USE BancoDados
GO
CREATE PROCEDURE Busca --- Declarando o nome da procedure
@CampoBusca VARCHAR (20) --- Declarando variável (note que utilizamos o @
antes do nome da variável)
AS
SELECT Codigo, Descrição --- Consulta
FROM NomeTabela
WHERE Descricao = @CampoBusca --- Utilizando variável como filtro para a
consulta
```

Para executar uma procedure basta utilizar a cláusula EXECUTE seguido pelo nome da procedure e na frente o valor a ser utilizado como parâmetro Exemplo:

```
EXECUTE Busca 'VALOR DO PARAMETRO';
```

Para excluir um procedure é necessário utilizar a cláusula DROP PROCEDURE como no exemplo abaixo.

```
DROP PROCEDURE Busca;
```