

Melhorar a qualidade do código Stored Procedures PL/SQL

Charles Boller

Faculdades Integradas Camões
{cboller79@gmail.com}

Abstract. *This article has a principle of indoctrinating new employees and a new standard of good practice in the language of stored procedures.*

Resumo. *Este artigo tem a intenção de doutrinar novos funcionários e colaboradores a seguirem padrões de boas práticas em linguagem de stored procedures.*

1. Introdução

Em um sistema que possui várias rotinas de processamento de alta performance desenvolvidas diretamente em servidores de banco de dados, sendo má utilizada se mostraram ineficazes no processamento das mesmas, em se tratando de performance.

As stored procedures, por serem executadas dentro do banco de dados, é a forma mais rápida de se processar uma rotina.

Doravante serão listados os padrões que devem ser seguidos para a escrita de stored procedures.

Tabela 1. Condições de padronização

Tipo	Descrição
Nome da procedure	Deve estar no infinitivo e indicar corretamente o que se está executando. O nome da procedure deve ser CamelCase. Ex.: Processar, Excluir.
Nome de function	Deve ter um substantivo no seu nome. O nome da function deve ser CamelCase. Ex: EhReferenciado, HostName, EhCPFValido.
NameSpace	Deve ser utilizado em conjunto com o namespace, para criar o contexto correto do nome. O caminho do namespace deve ser todo em minúsculo. Ex.: \$projeto\$.beneficiario.cadastro.AjustarMigracoes, \$projeto\$.prestador.preco.Calcular.pl2
Os parâmetros de declaração devem estar	Parâmetros de entrada IN; Parâmetros de entrada e saída INOUT;

na seguinte ordem:	<p>Parâmetros de saída OUT.</p> <p>Parâmetros INOUT devem ser evitados;</p>
Nomenclatura de parâmetro de entrada/saída	<p>O nome deve ser CamelCase, isto é, a primeira letra de cada palavra deve ser maiúscula, o nome deve iniciar com “p_”.</p> <p>Ex: p_FaturaHandle, p_DataInicial, p_QuantidadeDevolvida</p>
Nomenclatura de variáveis internas	<p>O nome deve ser CamelCase, isto é, a primeira letra de cada palavra deve ser maiúscula, o nome deve iniciar com “v_”.</p> <p>Ex: v_FaturaHandle, v_DataInicial, v_QuantidadeDevolvida.</p>
Nomenclatura de cursores	<p>O nome deve ser CamelCase, isto é, a primeira letra de cada palavra deve ser maiúscula, o nome deve iniciar com “c_”.</p> <p>Ex: c_Faturas, c_Guias, c_Benefs.</p>
Comentários	<p>Utilizar o comando “rem” para indicar no início da procedure, o que a procedure faz. Este comentário será mostrado durante a programação, ao se referenciar esta procedure.</p>
Performance	<p>Escrever as regras de negócio do sistema em stored procedures. Deixar para escrever fora das procedures, apenas o que não for possível ser feito em procedures.</p> <p>Querys com joins em tabelas muito grandes devem ser quebradas em querys menores.</p>
Boas práticas	<p>Evitar procedures com muitos parâmetros;</p> <p>Atribuir NULL para as variáveis de um SELECT, antes da execução deste, quando houver a possibilidade do comando não retornar nenhum registro;</p> <p>Validar a possibilidade de reutilizar uma procedure antes de criar uma nova.</p>

2. 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.

Em geral, empresas que exigem grande nível de segurança junto a quantidade massiva de dados, utilizam uma stored procedure para muitas operações em comum. Os procedimentos podem assegurar que as operações sejam registradas de forma correta, segura e rápida.

3. Por que é mais seguro?

Seguindo a linha de raciocínio, 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.

Referências

Boas práticas para criar e versionar procedures: Disponível em:

<https://pt.stackoverflow.com/questions/21149/boas-pr%C3%A1ticas-para-criar-e-versionar-procedures>.

Acesso em: 05 agosto. 2018.