**CENTRO PAULA SOUZA**

**ETEC IRMÃ AGOSTINA**

**Técnico em Desenvolvimento de Sistemas**

**Cesar de Sant’Ana Ng 20142**

**Thiago Torrecilha Lopes da Silva 20098**

**Procedimento armazenado e suas funcionalidades**

**São Paulo**

**2021**

**Cesar de Sant’Ana Ng 20142**

**Thiago Torrecilha Lopes da Silva 20098**

**Procedimento armazenado e suas funcionalidades**

**Trabalho introdutório aos Stored Procecures apresentado ao Curso Técnico em Desenvolvimento de sistemas da Etec Irmã Agostina, orientado pelo Professor Ismael Augusto Vasconcelos na matéria de Banco de Dados lll, com data de entrega final de 10 de julho de 2021.**

**São Paulo**

**2021**

Procedimento armazenado ou Stored Procedure é uma coleção de comandos em SQL, que podem ser executadas em um Banco de dados de uma só vez, como em uma função. Os procedimentos armazenados encapsulam tarefas repetitivas, aceitam parâmetros de entrada, são capazes de utilizar os comandos como IF e ELSE, WHILE, LOOP, REPEAT e CASE, além de poderem chamar outros procedimentos armazenados e retornam um valor de status (para indicar aceitação ou falha na execução).

Existem diversos usos para procedimentos armazenados, pois, dentro do procedimento podemos utilizar diversos tipos de comandos como INSERT, UPGRADE, DELETE, MERGE, DROP, CREATE e ALTER assim fornecendo um grande leque de utilidades para procedimentos armazenados.

Um procedimento armazenado também pode ser utilizado para validação de dados e controle de acesso.

Os procedimentos são como funções que serão guardadas no servidor, que podem ou não ser executadas através de um comando “EXEC [nome da procedure]” (em seu caso sem a necessidade de parâmetros de entrada).

Por ser executada dentro do servidor, o tráfego de dados existente na rede é drasticamente reduzido, pois, as únicas coisas que serão passadas pela rede são os valores dos parâmetros de entrada e o nome do procedimento assim otimizando o tempo de execução, diminuindo o uso da CPU e diminuindo a necessidade de memória. Além criar mecanismos de segurança entre a manipulação dos dados do Banco de Dados. Exemplo: (MS-SQL Server).

Dentro de uma Stored Procedure, você pode escrever comandos DML (Data Manipulation Language = Insert, Delete, Update, Merge) e DDL (Data Definition Language – Create, Drop, Alter). Ou seja, você pode fazer muita coisa!

O SQL Server armazena em cache os planos de execução das consultas e, normalmente, os reutiliza em execuções subsequentes da procedure para economizar tempo, CPU e recursos de memória associados à otimização das consultas.

As Stored Procedures fornecem muitos benefícios em comparação à implementação da lógica de negócios no aplicativo. Eles encapsulam a lógica para permitir a reutilização e ocultam a complexidade. É muito mais fácil aplicar alterações a um procedimento armazenado com um comando ALTER PROCEDURE simples em comparação à implementação de alterações no aplicativo.

Além disso, com os procedimentos armazenados, você precisa ter menos tráfego de rede, pois quando você chama o procedimento armazenado a partir do aplicativo, tudo o que é passado pela rede é apenas o nome do procedimento e seus parâmetros. O fluxo é executado na engine do banco de dados e, em seguida, apenas o resultado final é enviado para o aplicativo.

Quando você implementa a lógica no aplicativo, geralmente são necessárias mais “viagens” entre o aplicativo e o banco de dados e, consequentemente, mais tráfego de rede. As Stored Procedures também simplificam o tratamento de segurança no banco de dados. Muitas vezes, você não deseja conceder aos usuários permissões para consultar diretamente e modificar dados nas tabelas. Em vez disso, você quer que eles consigam realizar essas tarefas apenas indiretamente por meio de procedimentos armazenados. Para conseguir isso, conceda aos usuários permissões EXECUTE na Stored Procedure, sem conceder acesso direto aos objetos.

Referencias: <https://pt.wikipedia.org/wiki/Procedimento_armazenado>

Referencias: https://imasters.com.br/banco-de-dados/afinal-de-contas-o-que-e-uma-stored-procedure