

Aplicações com Banco de Dados

21/08/2024

Conteúdo da aula

- Stored Procedure

Stored Procedure

Conjunto de comandos T- SQL que podem ser executados de uma só vez, como em uma função. Suportam parâmetros de entrada e saída, e que podem ou não retornar resultados.

Um Stored Procedure pode reduzir o tráfego na rede (quando o SP é chamado a partir do aplicativo, tudo o que é passado na rede é seu nome e seus parâmetros), melhorar a performance de um BD (reutilização de planos de execução), criar tarefas agendadas, etc.

Stored Procedure

Também é possível simplificar o tratamento de segurança no BD, pois muitas vezes não é viável liberar permissão de consulta diretamente à várias tabelas, é possível liberar o acesso indiretamente pelos procedimentos.

Além da segurança, SP são muito utilizados quando aplicações escritas em várias linguagens ou que trabalhem em diferentes plataformas, precisam executar as mesmas operações de banco de dados.

Criando Stored Procedure

```
CREATE PROCEDURE <Procedure_Name >
-- Add the parameters for the stored procedure here
<@Param1, sysname, @p1> <Datatype_For_Param1, , int> =
<Default_Value_For_Param1, , 0>,
<@Param2, sysname, @p2> <Datatype_For_Param2, , int> =
<Default_Value_For_Param2, , 0>
AS
BEGIN
SET NOCOUNT ON;
SELECT <@Param1, sysname, @p1>, <@Param2, sysname, @p2>
END
```

Criando Stored Procedure

```
create procedure nome_procedure  
as  
Begin  
    select * from Clientes  
End
```

Executando Stored Procedure

Para executar uma procedure, utilize a cláusula EXECUTE (EXEC), seguido pelo nome da procedure e na frente o nome a ser passado como parâmetro (quando existir)

```
exec nome_procedure
```

Alterando Stored Procedure

Para alterar uma procedure, é utilizado ALTER PROCEDURE

alter procedure nome_procedure
<bloco de código a ser alterado>

```
alter procedure nome_procedure  
as  
Begin  
    select * from clientes where idtipo=2  
End
```


Excluindo Stored Procedure

Para alterar uma procedure, é utilizado DROP PROCEDURE

```
drop procedure nome_procedure
```

```
drop procedure nome_procedure
```

Utilizando parâmetros

- **Parâmetro de entrada**: permite que o comando passe um valor de dados para SP

```
create procedure nome_procedure (@tipo_cliente as int)
as
Begin
    select * from Clientes where idTipo=@tipo_cliente
End

exec nome_procedure 1
```

Utilizando parâmetros

- **Parâmetro de saída**: permite que o SP passe um valor de dados ou uma variável de cursor de volta para o comando

```
create procedure nome_procedure3 (@cod_cliente as varchar(10),
@nome_cliente as varchar(50) output)
as
Begin
    select @nome_cliente= NomeContato from Clientes where
    idcliente=@cod_cliente
End

declare @nomeCliente as varchar(50)
exec nome_procedure3 'ALFKI', @nomeCliente output
print @nomeCliente
```

Visualizar o conteúdo de SP

Para visualizar o conteúdo de um Stored Procedure existente, é preciso utilizar outro SP, o `sp_helpdesk`

```
exec sp_helptext nome_procedure3
```

Criando SP criptografado

```
create procedure sp_teste (@cod_cliente as varchar(10),  
@nome_cliente as varchar(50) output)  
with encryption  
as  
Begin  
    select @nome_cliente= NomeContato from Clientes where  
    idcliente=@cod_cliente  
End
```

Exercícios

- 1) Crie uma Stored Procedure para listar o nome e a idade de todos os clientes
- 2) Altere a SP do exercício 1, para ser possível passar o idfuncionario como parâmetro, para selecionar apenas um cliente
- 3) Crie uma SP que envolva 3 tabelas, e que passe 2 parâmetros de entrada

(Banco de Dados Aventuras)

Referências

ELMASRI, Ramez. Sistemas de banco de dados. Tradução Marília Guimarães Pinheiro. 4. ed. São Paulo: Pearson Addison Wesley, 2006.