

Você está em

DevMedia

Artigo

Procedures e funções no MySQL

Veja neste artigo como criar, alterar, excluir e executar procedures e funções no MySQL



Marcado como lido



Anotar

Artigos



Banco de Dados



Procedures e funções no MySQL

```
1 CREATE PROCEDURE sp_name ([parameter[,...]])  
2 [characteristic ...] routine_body  
3  
4 CREATE FUNCTION sp_name ([parameter[,...]])  
5 [RETURNS type]  
6 [characteristic ...] routine_body  
7  
8 parameter:
```



Você está em

DevMedia

```
15 | LANGUAGE SQL
16 | | [NOT] DETERMINISTIC
17 | | SQL SECURITY {DEFINER | INVOKER}
18 | | COMMENT string
19 |
20 | routine_body:
21 | Valid SQL procedure statement(s)
```

Listagem 1. Criar procedures e funções no MySQL

A cláusula `RETURNS` pode ser especificada apenas por uma `FUNCTION`. É usada para indicar o tipo de retorno da função, e o corpo da função deve conter uma instrução `RETURN value`.

A lista de parâmetros entre parênteses deve estar sempre presente. Se não houver parâmetros, uma lista de parâmetros vazia de `()` deve ser usada. Cada parâmetro é um parâmetro `IN` por padrão. Para especificar outro tipo de parâmetro, use a palavra chave `OUT` ou `INOUT` antes do nome do parâmetro. Especificar `IN`, `OUT` ou `INOUT` só é válido para uma `PROCEDURE`.

A instrução `CREATE FUNCTION` é usada em versão novas do MySQL para suporte a UDFs (User Defined Functions - Funções Definidas pelo Usuário). As UDFs continuam a ser suportadas, mesmo com a existencia de stored functions. Uma UDF pode ser considerada como uma stored function externa. No entanto, note que stored functions compartilham os seus namespace com as UDFs.

Um framework para stored procedures externas serão introduzidas em um futuro próxima. Isto permitirá que você escreva stored procedures em outras linguagens além de SQL. Provavelmente, uma das primeiras linguagens a ser suportada sea

Você está em

DevMedia

Uma função é considerada "determinística" se ela sempre retorna o mesmo resultado para os mesmos parâmetros de entrada, e "não determinística" caso contrário. O otimizador pode usar este fato. Atualmente, a característica `DETERMINISTIC` é aceita, mas ainda não é usada.

A característica `SQL SECURITY` pode ser usada para especificar se a rotina deve ser executada usando as permissões do usuário que criou a rotina, ou o usuário que a chamou. O valor padrão é `DEFINER`. Este recurso é novo no `SQL:2003`.

O MySQL ainda não usa o privilégio `GRANT EXECUTE`. Assim, por enquanto, se um procedimento `p1()` chama a tabela `t1`, o usuário deve ter privilégios na tabela `t1` para chamar o procedimento `p1()` com sucesso.

```
1 | MySQL stores the SQL_MODE settings in effect at the time a routine is  
2 | and will always execute routines with these settings in force.
```

A cláusula `COMMENT` é uma extensão do MySQL, e pode ser usada para descrever o stored procedure. Esta informação é exibida pelas instruções `SHOW CREATE PROCEDURE` e `SHOW CREATE FUNCTION`.

O MySQL permite rotinas contendo instruções DDL (como `CREATE` e `DROP`) e instruções de transação SQL (como `COMMIT`). Isto não é exigido por padrão e depende de especificações de implementação.

Você está em

DevMedia

A seguir temos um exemplo de uma `stored procedure` simples que usa um parâmetro `OUT`. O exemplo usa o comando `delimiter` do cliente `mysql` para alterar o delimitador de instrução para antes da definição do procedure. Isto permite que o delimitador `;` usado no corpo de procedure seja passado para o servidor em vez de ser interpretado pelo `mysql`.

```
1  mysql> delimiter |
2
3  mysql> CREATE PROCEDURE simpleproc (OUT param1 INT)
4      -> BEGIN
5      ->   SELECT COUNT(*) INTO param1 FROM t;
6      -> END
7      -> |
8  Query OK, 0 rows affected (0.00 sec)
9
10 mysql> CALL simpleproc(@a)|
11 Query OK, 0 rows affected (0.00 sec)
12
13 mysql> SELECT @a|
14 +-----+
15 | @a    |
16 +-----+
17 | 3     |
18 +-----+
19 1 row in set (0.00 sec)
```

A seguir esta um exemplo de uma função que utiliza um parametro, realiza uma operação usando uma função SQL e retorna o resultado:

Você está em

DevMedia

```
3 | query OK, 0 rows affected (0.00 sec) mysql> SELECT hello( 'world' );
4 | +-----+ | hello('world') |
5 | +-----+ |
6 | Hello, world! |
7 | +-----+ 1 row in set (0.00 sec)
8 |
9 | ALTERAR PROCEDURE E FUNCTION
10 |
11 | ALTER PROCEDURE | FUNCTION sp_name [characteristic ...]
12 |
13 | characteristic:
14 |   NAME newname
15 |   | SQL SECURITY {DEFINER | INVOKER}
16 |   | COMMENT string
```

Este comando pode ser usado para renomear uma stored procedure ou function, e para alterar suas características. Mais de uma mudança pode ser especificada em uma instrução `ALTER PROCEDURE` ou `ALTER FUNCTION`.

```
1 | EXCLUIR PROCEDURE E FUNCTION
2 | DROP PROCEDURE | FUNCTION [IF EXISTS] sp_name
```

Este comando é usado para deletar uma stored procedure ou function. Isto é, a rotina especificada é removida do servidor. A cláusula `IF EXISTS` é uma extensão do MySQL. Ela previne que um erro ocorra se o procedimento ou função não existe. Um aviso é produzido e pode ser visualizado com `SHOW WARNINGS`.

Você está em

DevMedia

Marcado como lido



Anotar



Por Devmedia

Em 2006

Suporte ao aluno - Tire a sua dúvida.

[Planos de estudo](#)[Fale conosco](#)[Plano para Instituição de ensino](#)[Assinatura para empresas](#)[Assine agora](#)

Hospedagem web por Porta 80 Web Hosting



6



