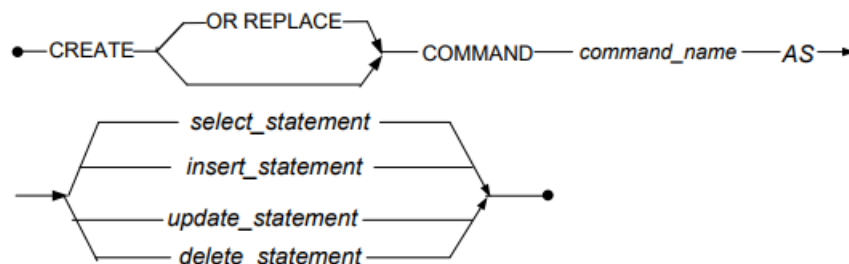


11. Stored Commands

Um stored commands é uma instrução SQL DML compilada e armazenada no banco de dados. Um stored commands é pré-compilado em um formato executável, permitindo que o mesmo comando seja executado sem a necessidade de compilação e otimização. É possível criar um stored commands para qualquer instrução SQL frequentemente utilizada, obtendo assim um melhor desempenho. Stored commands são considerados um subconjunto de stored procedures que contêm apenas uma instrução SQL, sem lógica de programação.

11.1 Creating Stored Commands

Use a instrução CREATE COMMAND para criar um stored commands.



OR REPLACE é utilizado para recriar um stored commands que já existe, ou seja, os usuários podem usar essa cláusula para alterar a definição de um stored commands existente. Parâmetros de entrada na instrução SQL podem ser usados ao criar um stored commands.

O valor real dos parâmetros de entrada para um stored commands pode ser atribuído no momento da execução.

Exemplo 1:

Para criar um stored commands chamado `sc_student_insert` para a instrução SQL DML utilizando uma tabela com a definição `tb_student` (`id` INT, `score` INT, `name` CHAR(32)):

```
dmSQL> INSERT INTO tb_student VALUES (1, ?, ?);
```

Exemplo 2:

Alternativamente, use CREATE COMMAND:

```
dmSQL> CREATE COMMAND sc_student_insert AS INSERT INTO tb_student
VALUES
(1, ?, ?);
```

Exemplo 3:

Alternativamente, use CREATE OR REPLACE COMMAND:

```
dmSQL> CREATE OR REPLACE COMMAND sc_student_insert AS INSERT INTO
tb_student
VALUES (1, ?, ?);
```

Exemplo 4:

Para criar stored commands para outras instruções DML:

```
dmSQL> CREATE COMMAND sc_student_select AS SELECT id,name FROM
tb_student;
dmSQL> CREATE COMMAND sc_student_update AS UPDATE tb_student SET id
= id+1 WHERE
score > ?;
dmSQL> CREATE COMMAND sc_student_delete AS DELETE FROM tb_student
WHERE score
> ?;
```

Após criar um stored commands, um usuário com permissão pode executá-lo diretamente usando dmSQL ou em um programa de aplicação. Se o stored commands tiver parâmetros de entrada, seu valor pode ser determinado usando marcas de parâmetros, constantes, NULL, DEFAULT ou funções internas (funções internas não podem ter argumentos) ao executar o stored commands. Ao executar stored commands, o número de parâmetros de entrada do stored commands deve ser igual ao número de parâmetros fornecidos.

11.2 Executing a Stored Command

Use a instrução EXECUTE COMMAND para executar um stored commands.

Exemplo 1:

```
dmSQL> EXECUTE COMMAND sc_student_insert (200, 'john');
```

Exemplo 2:

```
dmSQL> EXECUTE COMMAND sc_student_insert (DEFAULT, ?);
```

Exemplo 3:

```
dmSQL> EXECUTE COMMAND sc_student_insert (?, NULL);
```

Exemplo 4:

```
dmSQL> EXECUTE COMMAND sc_student_insert (?, ?);
```

Um stored commands pode ser removido quando não for mais necessário.

11.3 Rebuilding a Stored Command

Use a instrução REBUILD COMMAND para reconstruir um stored commands

• — REBUILD COMMAND — *command_name* — •

Exemplo:

```
dmSQL> REBUILD COMMAND sc_student_insert;
```

11.4 Dropping a Stored Command

Use a instrução DROP COMMAND para remover um stored commands

• — DROP COMMAND — IF EXISTS — *command_name* — •

Exemplo 1:

```
dmSQL> DROP COMMAND sc_student_insert;
```

Exemplo 2:

```
dmSQL> DROP COMMAND IF EXISTS sc_student_insert;
```

11.5 Stored Command Security

Os stored commands são tratados como objetos de esquema do banco de dados. Portanto, os usuários devem considerar a segurança e os privilégios de objeto ao criá-los ou usá-los.

Somente o criador ou usuários com o privilégio RESOURCE podem criar um stored commands. Um usuário só pode criar um comando armazenado a partir de uma instrução SQL DML se tiver privilégios para executar a instrução SQL DML.

Exemplo:

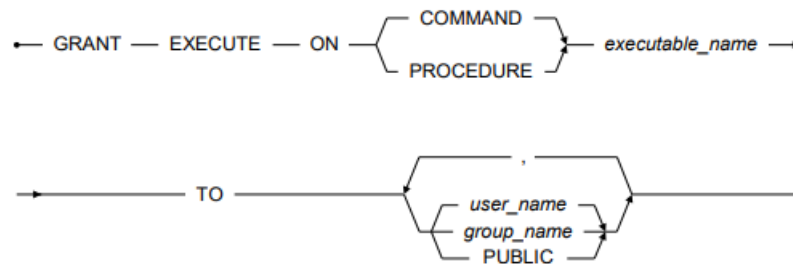
O usuário joe, com o privilégio RESOURCE, deseja criar um stored commands sc_CheckDate com a seguinte sintaxe:

```
dmSQL> CREATE COMMAND sc_CheckDate AS SELECT FirstName, LastName,  
Hiredate FROM  
SYSADM.tb_staff WHERE HireDate > '1995-01-01';
```

A tabela tb_staff é propriedade de SYSADM, portanto, o administrador do sistema deve primeiro conceder permissão de seleção ao usuário joe na tabela SYSADM.tb_staff antes que o usuário joe possa criar o stored commands.

Um usuário deve ter o privilégio de DBA para um comando armazenado para executá-lo. Para permitir que um stored commands seja usado por outros, o usuário pode conceder o privilégio de execução para o stored commands. No entanto, somente usuários com os privilégios necessários (DBA, SYSDBA, SYSADM, o criador do stored commands ou outros a quem o privilégio foi concedido) podem conceder ou revogar privilégios de execução para stored commands.

Granting Execute Privilege

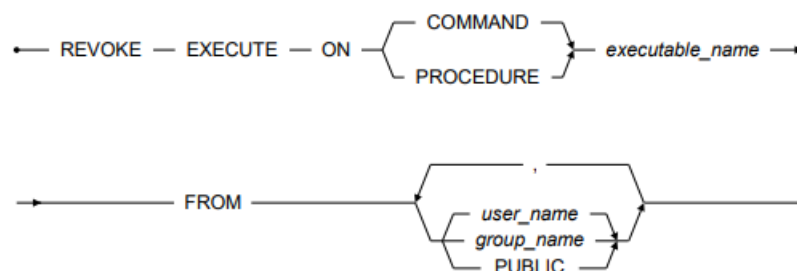


Exemplo:

Para conceder o privilégio EXECUTE ao usuário John para o comando `sc_student_insert`:

```
dmSQL> GRANT EXECUTE ON COMMAND sc_student_insert TO John;
```

Revoking Execute Privileges



Exemplo:

Para revogar o privilégio EXECUTE de John para o comando `sc_student_insert`:

```
dmSQL> REVOKE EXECUTE ON COMMAND sc_student_insert FROM John;
```

11.6 Lifecycle of a Stored Command

Um stored commands se tornará inválido se uma das tabelas relacionadas no stored commands for removida ou alterada. Se algum programa foi escrito anteriormente usando informações antigas de colunas, isso pode causar resultados imprevisíveis no momento da execução.

O benefício do stored commands é a melhoria do desempenho ao executar repetidamente um comando SQL. O DBMaker também considera quando o plano de execução deve ser atualizado, como no caso do comando `UPDATE STATISTICS`.

Quando o comando UPDATE STATISTICS é emitido, todos os planos de execução para o stored commands serão atualizados para alcançar um melhor desempenho.

11.7 Getting Information for Stored Commands

Informações sobre stored commands podem ser encontradas na tabela de sistema SYSCMDINFO. A Tabela 11-1 lista as colunas da tabela SYSCMDINFO e seus valores.

COLUMN NAME	VALUE	COMMENT
MODULENAME	Nome do módulo ao qual o comando armazenado pertence	Esta coluna é usada por uma aplicação ESQL ou por uma procedure armazenada. Pode ser ignorada se for um comando armazenado puro.
CMDNAME	Nome do comando armazenado	None
CMDOWNER	Proprietário do comando armazenado	None
STATEMENT	String SQL original	None
NUM_PARM	Número de parâmetros	None
STATUS	0, 1 ou 2	0 - Comando armazenado inválido. Não pode ser executado. 1 - Comando armazenado válido. Pode ser executado. 2 - O comando armazenado precisa ser reatribuído. Pode ser executado após a reatribuição interna.
REBTIME	Hora de reconstrução do comando armazenado	None
CMDPLAN	Exportar string do plano de execução do comando armazenado	None

Para recuperar informações sobre stored commands usando a seguinte instrução em dmSQL:

```
dmSQL> SELECT * FROM SYSCMDINFO;
```