

```

mysql> CALL sp_demite(24);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM cadfun;

```

CODFUN	NOME	DEPTO	FUNCAO	SALARIO	ADMISSAO	FILHOS	CPF
1	SILVIO DE MENESES	2	ANALISTA	2000.00	2006-08-10	1	10020011199
6	SILVIA DA SILVA	5	ANALISTA	2000.00	2006-08-10	2	10020022299
8	ANTONIO DA SILVA	5	GERENTE	2184.33	2006-07-05	2	10020033399
10	ANTONIA DE CARVALHO	5	SECRETARIA	2530.62	2006-07-07	4	10020044499
11	SILVANA DOS SANTOS	4	VENDEDOR	1683.00	2006-07-07	0	10020055599
12	ANTONIO DOS SANTOS	4	PROGRAMADOR	1155.00	2006-10-10	1	10020066699
20	AUDREY TOLEDO	2	SUPERVISOR	1700.00	2006-07-05	4	10020077799
21	EPHANIMONDAS DA SILVA	4	PROGRAMADOR	1155.00	2006-10-10	2	10020088899
22	SANDRA RANZANO	2	ANALISTA	2000.00	2006-07-01	0	1002211199
50	MARCOS INACIO	2	GERENTE	2184.33	2006-09-25	2	11122233344

```

10 rows in set (0.00 sec)

mysql> SELECT * FROM morto;

```

CODFUN	NOME	DEPTO	FUNCAO	SALARIO	ADMISSAO	FILHOS	CPF
2	WILSON DE MACEDO	3	PROGRAMADOR	1155.00	2006-01-15	1	NULL
3	ANA BASTOS	3	VENDEDOR	1683.00	1999-10-21	3	NULL
4	CARLOS BASTOS	5	VENDEDOR	1683.00	2004-10-21	0	NULL
5	AUGUSTO SOUZA	3	PROGRAMADOR	1155.00	2006-04-26	2	NULL
7	APARECIDA SILVA	3	SECRETARIA	2530.62	1980-05-10	0	NULL
9	SOLANGE PROENÇA	5	SUPERVISOR	1759.46	1999-12-15	1	NULL
12	CARLOS ALBERTO	3	VENDEDOR	1683.00	2000-12-21	NULL	NULL
15	MARCOS HENRIQUE	2	GERENTE	2184.33	2006-05-25	0	NULL
21	EPHANIMONDAS DA SILVA	4	PROGRAMADOR	1155.00	2006-10-10	2	NULL
24	MARCIO CARNUITO	2	PROGRAMADOR	1200.00	2006-07-10	0	10022233399
25	PEDRO SILVA	3	SUPERVISOR	1759.46	2000-10-21	3	NULL

```

11 rows in set (0.00 sec)

mysql>

```

Figura 9.18 - Criação e execução da sub-rotina "sp_demite()" e seu resultado.

Com o último exemplo observe como é possível usar *stored procedures* (procedimentos armazenados). Muitas das trabalhosas tarefas a serem executadas com as tabelas de um banco de dados podem ser facilmente automatizadas.

Assim como ocorre em outras linguagens, a linguagem de *script* utilizada para o desenvolvimento de procedimentos armazenados opera com escopo de variáveis.

O escopo de uma variável está relacionado à sua visibilidade dentro do código. Por exemplo, as variáveis iniciadas com o caractere arroba (@) são externas à sub-rotina, ou seja, são variáveis com escopo global, já as definidas dentro da sub-rotina com o comando DECLARE são internas, ou seja, com escopo local. No entanto, é possível estabelecer escopos globais e locais de variáveis locais dentro do código de uma sub-rotina. Considere o exemplo seguinte:

```

DELIMITER $
CREATE PROCEDURE sp_escopo()
BEGIN
    DECLARE Palavra VARCHAR(6) DEFAULT 'Fora';
    BEGIN
        DECLARE Palavra VARCHAR(6) DEFAULT 'Dentro';
        SELECT Palavra;
    END;
    SELECT Palavra;
END$
DELIMITER ;

CALL sp_escopo();

```

A Figura 9.19 exibe a sub-rotina **sp_escopo()** com os resultados **Palavra - Dentro** e **Palavra - Fora**. A variável Palavra apresenta em dois momentos valores diferentes. Isso ocorre devido ao escopo de sua definição.