```
DELIMITER $
CREATE PROCEDURE sp_fat4(IN Valor INTEGER)
BEGIN

DECLARE Fator INTEGER DEFAULT 1;
DECLARE I INTEGER DEFAULT 1;
SET Fator = 1;
SET I = 1;
calculo: REPEAT
SET Fator = Fator * I;
SET I = I + 1;
UNTIL (I > Valor) END REPEAT calculo;
SELECT Fator;
END$
DELIMITER;

CALL sp_fat4(5);
```

A Figura 9.16 mostra a sub-rotina **sp\_fat4()** com o resultado **120** que é o valor da fatorial de **5**, ou seja, **5!** = **120**. Observe o uso do comando DEFAULT que determina o valor padrão inicial **1** para as variáveis I e Fator.

```
Delimiter:

mysql) Delimiter:

mysql) Call sp_fat3(5);

i Fator i

i row in set (0.00 sec)

Guery OK. 0 rows affected (0.01 sec)

mysql) Cellmiter S

mysql) Cellmiter S

Declare Fator integer DeFault 1;

Declare I integer DeFault 1;

Declare I integer DeFault 1;

SET Fator = 1;

Calculo: REPEAT

SET Fator = 1;

SET Fator = 1;

MIL (1 where) END REPEAT calculo;

SELECT Fator;

Pusql) Call sp_fat4(5);

Fator

I row in set (0.00 sec)

query OK, 0 rows affected (0.00 sec)

mysql) Call sp_fat4(5);

Fator

I row in set (0.00 sec)

query OK, 0 rows affected (0.00 sec)

mysql) Call sp_fat4(5);

Fator

I row in set (0.00 sec)

query OK, 0 rows affected (0.00 sec)

mysql) Call sp_fat4(5);
```

Figura 9.16 - Criação e execução da stored procedure "sp\_fat4()".

Após alguns exemplos de construção de *procedures* aparece uma sub-rotina de registro de demissão. Lembre-se de que, antes de demitir um funcionário da tabela **cadfun**, precisa ser cadastrado na tabela de arquivo **morto**. Antes de definir a sub-rotina que fará o trabalho de inserção de um registro na tabela **morto** e a remoção do mesmo registro da tabela **cadfun**, utilize os comandos:

```
SELECT * FROM cadfun;
SELECT * FROM morto;
```

Para ver o estado atual de ambas as tabelas, como mostra a Figura 9.17. Para melhor visualização dos dados em tela, com o ponteiro do *mouse* voltado para as bordas laterais da janela arraste a borda no sentido de aumentar a área de visualização.