

## ABAPZombie Guide to ABAP – Parte 35 – DESCRIBE TABLE

20/08/2012 07:00

O DESCRIBE TABLE retorna o tamanho de uma tabela, simples assim.

```
REPORT zombie_describe_table.

SELECT *
FROM scarr
INTO TABLE t_scarr.

DESCRIBE TABLE t_scarr.
DESCRIBE TABLE t_scarr KIND v_tipo.
DESCRIBE TABLE t_scarr LINES v_linhas.
DESCRIBE TABLE t_scarr OCCURS v_tamanho_inicial.

WRITE:
/'sy-tfill'      , sy-tfill      , "quantidade de linhas
/'sy-tleng'      , sy-tleng      , "tamanho em bytes de uma linha
/'v_tipo'        , v_tipo        , "tipo da tabela: "T" standard table, "S" sorted table e
/'v_linhas'      , v_linhas      , "quantidade de linhas
/'v_tamanho_inicial' , v_tamanho_inicial. "tamanho inicial da tabela (ou occurs)
```

Você pode usar os parâmetros opcionais KIND, LINES e OCCURS, para retornar o tipo da tabela ("T" standard table, "S" sorted table e "H" hashed tables) , o número de linhas e o tamanho inicial respectivamente. Além da variável solicitada, o sistema também atualiza as variáveis de sistema SY-TFILL e SY-TLENG. Caso não seja solicitado nenhum, somente as variáveis de sistema serão atualizadas.

Pronto agora você não precisa mais criar aquela variável safada v\_lines, basta usar o sy-tfill 😊

### Comentários

**Custodio** — 20/08/2012 23:01

Comentei ontem aqui e sumiu :S

Enfim, vou repetir...

Provavelmente voces ja tem planejad de comentar isso, mas esse describe para descobrir o numero de linhas esta morto, se nao me engano desde a 7.00. Muito mais maroto eh usar:

```
v_linhas = lines( t_scarr ).
ou ainda:
```

```
if lines( t_scarr ) > 0.
*do something...
endif.
```

abracos.

**Fawcs** — 21/08/2012 09:10

legal é usar isso no meio do debug =D. Coloca um watchpoint com condição `lines( t_scarr ) > 100`, já tive que usar algumas vezes!