

# Comandos Auxiliares

Laboratório de Bases de Dados

Profa. Dra. Cristina Dutra de Aguiar Ciferri

Monitor: Marcus Vinícius Custódio

# TO\_CHAR

TO\_CHAR (data [,formato\_char])  
ou  
TO\_CHAR (número [,formato\_char])

- Converte um valor tipo data ou número para um valor char.

# TO\_DATE

TO\_DATE (char [,formato\_char])

- Converte uma expressão char para date.

# TO\_NUMBER

TO\_NUMBER (char)

- Converte para numérico um valor do tipo char.

# INITCAP

INITCAP (char)

- Retorna a inicial de cada palavra em letra maiúscula e as demais letras em minúsculo.

# LOWER

LOWER (char)

- Retorna todas as letras de todas as palavras que compõem o argumento em letras minúsculas.

# UPPER

UPPER (char)

- Retorna todas as letras de todas as palavras que compõem o argumento em letras maiúsculas.

# LPAD

LPAD (char\_val\_original, tamanho [, char\_preenchimento])

- Retorna a expressão char\_val\_original, preenchida à esquerda com char\_preenchimento o número de vezes necessárias para completar o tamanho estabelecido.

# RPAD

RPAD (char\_val\_original, tamanho [, char\_preenchimento])

- Retorna a expressão char\_val\_original, preenchida à direita com char\_preenchimento o número de vezes necessárias para completar o tamanho estabelecido.

# RTRIM

RTRIM (char\_val\_original, [, conjunto\_caracteres])

- Retorna a expressão char\_val\_original, sem os valores à direita que estejam contidos em conjunto\_caracteres.

# SUBSTR

SUBSTR (char, posição\_inicial[,tamanho])

- Retorna uma parte do valor original (substring), ou seja, uma string cujo comprimento será determinado pelo parâmetro 'tamanho', extraído da string 'char' (coluna, expressão ou constante), a partir da posição expressa no parâmetro 'posição\_inicial'.

# INSTR

INSTR (char1, char2 [, posição\_inicial, ocorrência])

- Retorna a posição da enésima ocorrência de uma determinada string2 dentro da string1.

# REPLACE

REPLACE (char, texto\_antigo[,texto\_novo])

- Retorna a string com todas as ocorrências do texto antigo trocadas pelo novo.

# TRANSLATE

```
TRANSLATE (char, conjunto_caracteres_antigo,  
           conjunto_caracteres_novo)
```

- Retorna o valor original com todas as ocorrências dos elementos pertencentes ao conjunto de caracteres antigos trocadas pelos correspondentes no conjunto novo.

# LENGTH

LENGTH (char)

- Retorna o número total de caracteres existentes (comprimento da string).

# ABS

ABS (number)

- Retorna o valor absoluto do número passado como argumento.

# CEIL

CEIL (number)

- Retorna o número inteiro maior ou igual ao número do argumento.

# FLOOR

FLOOR(number)

- Retorna o maior número inteiro menor ou igual ao argumento.

# MOD

MOD (dividendo, divisor)

- Retorna o resto da divisão do dividendo pelo divisor.

# POWER

POWER (base, expoente)

- Retorna o valor da potenciação.

# SQRT

SQRT (number)

- Retorna a raiz quadrada do número.

# ROUND

ROUND (number, [casas\_decimais])

- Retorna o número arredondado em  $n$  casas decimais.

# TRUNC

TRUNC (number, [casas\_decimais])

- Retorna o número truncado em  $n$  casas decimais.

# SIGN

SIGN (number)

- Retorna:
  - 1 para valores maiores que zero;
  - 0 quando o valor for zero;
  - -1 quando o valor for menor que zero.

# SYSDATE

SYSDATE

- Retorna a data do sistema.

# LAST\_DAY

LAST\_DAY (data)

- Retorna o último dia do mês da data informada.

# NEXT\_DAY

`NEXT_DAY (date [,dia_da_semana_por_extenso])`

- Retorna a data correspondente ao próximo dia da semana informado.

# ADD\_MONTHS

ADD\_MONTHS (date, número)

- Retorna a data adicionada do número de meses especificado.

# TRUNC

TRUNC (data [,formato])

- Retorna a data correspondente ao início do período corrente.

# ROUND

ROUND (data [,formato])

- Retorna a data correspondente ao início do período se o formato for 'MM' e o dia estiver entre 1 e 15;
- Retorna o mês corrente e ao início do próximo mês se o dia estiver entre 16 e 31.

# MONTHS\_BETWEEN

MONTHS\_BETWEEN (data, data)

- Retorna o número de meses entre uma data e outra.

# EXTRACT

```
EXTRACT (parte_da_data FROM valor_data)
```

- Extrai e retorna o valor de uma parte específica de um campo de data.

# OPERAÇÃO DE DATAS EM DIAS

- Quando se quer a diferença (subtração) entre duas datas, o seu resultado é dado em número de dias, ou seja, o número de dias entra uma data e outra. Por exemplo:
  - $\text{Data1} - \text{Data2} = \text{Número Dias entre Data1 e Data2.}$

# OPERAÇÃO DE DATAS EM DIAS

- Ao subtrair ou adicionar um valor de uma data, esse valor subtraído ou adicionado representa um número de dias e o resultado será uma data, ou seja, a data resultante representa a data original adicionada ou subtraída de  $n$  dias.
- $\text{Data1} - \text{Número Dias} = \text{Data 2}$   
ou
- $\text{Data1} + \text{Número Dias} = \text{Data 2}$

# DECODE

```
DECODE (valor, comparação1, valor1 [,  
        comparaçãoN, valorN] [,valor_default])
```

- Retorna um valor decodificado de acordo com uma comparação. Caso o valor ou expressão do primeiro parâmetro seja igual à comparação 1, o valor 1 será usado; caso seja igual à comparação 2, o valor 2 será usado, e caso não seja igual a nenhuma das comparações, o valor\_default será usado.

# CASE

```
CASE [expressão] WHEN condição THEN valor  
[WHEN condição THEN valor]  
[ELSE valor]
```

- Possui funcionalidade muito semelhante à do DECODE, permitindo, no entanto, comparações diferentes da igualdade. Ou seja, pode-se utilizar qualquer comparação que se utilizaria em um comando IF, por exemplo.

# NULLIF

NULLIF (expressão1, expressão2)

- Compara duas expressões, retornando nulo se ambas forem iguais. Caso não sejam iguais, a primeira expressão será retornada.

# GREATEST

GREATEST (valor [,valor [...]])

- Retorna o maior valor de uma lista.

# LEAST

LEAST [valor [,valor [...]]])

- Retorna o menor valor de uma lista.

# NVL

NVL (valor, valor\_a\_ser\_retornado)

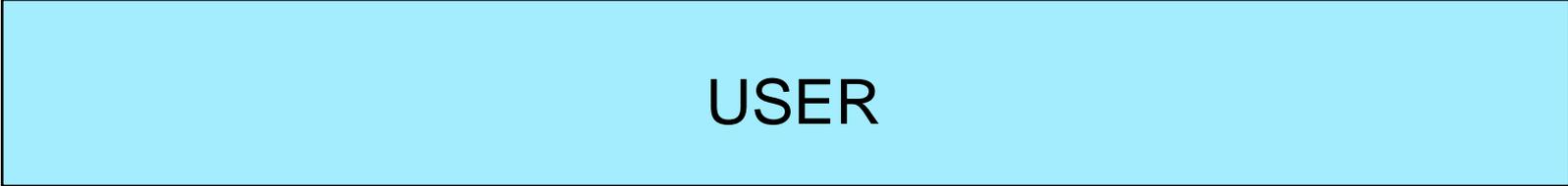
- Retorna o valor especificado pelo usuário caso o valor original seja nulo; do contrário, retorna o valor original.

# NVL2

NVL2 (valor1, valor2, valor3)

- Retorna o valor2 caso o valor1 não seja nulo, do contrário retorna o valor 3. O valor2 e o valor3 devem ser do mesmo tipo de dados, que não precisa ser o mesmo tipo de dados do valor1. Caso o valor2 e o valor3 sejam de tipos diferentes, o Oracle tentará converter o valor3 para o mesmo tipo do valor2.

# USER



USER

- Retorna o usuário corrente.