

[◀ VOLTAR](#)

Criação de relatórios utilizando filtros, operadores e funções em banco de dados

Criar relatórios a partir de dados existentes no Banco de dados utilizando comandos de seleção aplicando filtros, realizando cálculos, usando operadores e funções de banco de dados.

NESTE TÓPICO

[Marcar
tópico](#)

Em aulas anteriores, com o uso da criação de relatórios, vimos que estes podem ser muito simples, usando apenas a seleção, e outros podem ser mais complexos, dependendo da necessidade de pesquisa, das colunas envolvidas e dos procedimentos a serem adotados. Nesse contexto, já é conhecido o uso dos operadores que também fazem parte da criação de relatórios, bem como a realização de cálculos. Alguns deles são complexos – imagine ter que criar a fórmula para saber se um cliente tem como final de seu registro um número par ou ímpar ou então calcular a média salarial de uma tabela de funcionários, somar seus salários, saber se o conteúdo da coluna está ou não vazio e então contar a quantidade de salários para depois dividir... Difícil?

Não, nessa situação podemos usar recursos prontos, ou seja, fórmulas que podem ser processadas usando-se um nome e informando o que deve ser feito. Esses cálculos prontos são chamados de **função**, é o que aprenderemos nesta aula.

Funções numéricas

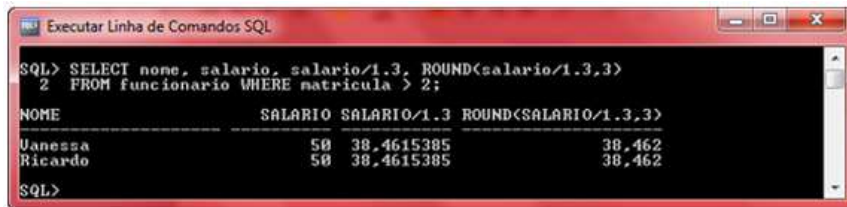
As funções numéricas são padrão SQL e só têm efeito sobre números ou colunas do tipo numérica - number.

Funções ROUND – arredonda valores ao mostrá-los.

Sintaxe: ROUND(nome da coluna, casas decimais)

Exemplo:

1. SELECT nome, salario, salario/1.3, ROUND(salario/1.3,3) FROM funcionario WHERE matricula > 2;



Executar Linha de Comandos SQL

```
SQL> SELECT nome, salario, salario/1.3, ROUND(salario/1.3,3)
2 FROM funcionario WHERE matricula > 2;
```

NOME	SALARIO	SALARIO/1.3	ROUND(SALARIO/1.3,3)
Vanessa	50	38,4615385	38,462
Ricardo	50	38,4615385	38,462

SQL>

O arredondamento acontece no processamento e, ao ser mostrado o resultado, serão exibidas três casas após a vírgula, como determinado na instrução.

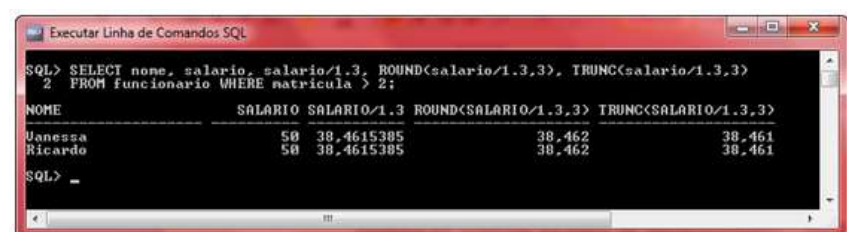
O padrão de arredondamento segue a análise da primeira casa após o limite de exibição; caso esta posição seja maior ou igual 5, é somado mais 1 à casa decimal limite. Observe o que aconteceu no relatório acima.

Função TRUNC ? mostra o número truncado. Diferentemente da função round, que realiza o arredondamento, esta função apenas exibe os valores até o limite imposto no comando, sem realizar o arredondamento.

Sintaxe: TRUNC(nome da coluna, casas decimais)

Exemplo:

1. SELECT nome, salario, salario/1.3, ROUND(salario/1.3,3), TRUNC(salario/1.3,2) FROM funcionario WHERE matricula > 2;



Executar Linha de Comandos SQL

```
SQL> SELECT nome, salario, salario/1.3, ROUND(salario/1.3,3), TRUNC(salario/1.3,3)
2 FROM funcionario WHERE matricula > 2;
```

NOME	SALARIO	SALARIO/1.3	ROUND(SALARIO/1.3,3)	TRUNC(SALARIO/1.3,3)
Vanessa	50	38,4615385	38,462	38,461
Ricardo	50	38,4615385	38,462	38,461

SQL>

Função MOD – determina o valor referente ao resto da divisão de um número.

Sintaxe: MOD(nome da coluna, divisor)

Exemplo:

1. SELECT nome, dt_admissao, matricula FROM funcionario WHERE MOD(matricula,2) = 0;

Esta instrução só vai exibir os dados existentes na seleção quando a matrícula do funcionário for um número par – o cálculo é realizado com a função MOD.

Cálculos com valor NULL

Para qualquer cálculo efetuado com um valor NULL – Vazio ou Nulo –, o resultado será sempre NULL. Portanto, para obter algum valor ou evitar divisão por um valor desconhecido, o que geraria um erro, utiliza-se a função NVL que transforma o valor NULL em um número ou *string* desejado.

Sintaxe:

NVL (expr1, expr2)

Onde:

Expr1 – Colunas que contêm valores nulos.

Expr2 – Valor em que os valores nulos serão transformados.

Exemplo:

1.

```
SELECT nome, salario, NVL(comissao,0) PORCENTAGEM, Salario * NVL(comissao,0) / 100 C  
OMISSAO FROM funcionario WHERE salario > 10;
```

Para a realização dessa instrução, algumas alterações foram feitas na tabela "funcionário"; a coluna "comissão" foi excluída e recriada novamente com mesmo nome e tipo de dados agora numérico com uma posição, os dados foram incluídos novamente, com os seguintes valores: para o funcionário de matrículas 1 e 3, a comissão incluída foi o número 1; para o de matrícula 2, o número 2; e, para o funcionário 4, não foi incluído nenhum valor, ficou nulo.

Funções com caracteres

As funções que trabalham caracteres têm grande importância no tratamento desses dados, uma vez que o responsável pelo sistema não se preocupa com a consistência dos dados. Os profissionais de banco de dados poderão ter dificuldades para gerar relatórios, por exemplo, como já foi visto, não existem problemas com relação à digitação dos comandos nos formatos maiúsculo e minúsculo, mas na consulta onde usaremos um dado carácter (char) ou uma *string* (varchar, varchar2) o problema existe, há diferença na pesquisa entre os dados maiúsculos e minúsculos. Nesse instante, para poupar aborrecimentos, podem-se usar essas funções que trabalham os tipos citados.

1.

```
SELECT nome, INITCAP(nome), UPPER(nome), LOWER(nome) FROM funcionario;
```



As funções caracter como todas as outras que já foram vistas também podem ser utilizadas nas condições, onde seu efeito é melhor. Imagine que seja necessário criar um relatório em que o resultado seja o nome de um funcionário. Parece simples, mas, se não existir um padrão de armazenamento, podemos demorar muito para construir a instrução, pois o caracter ou *string* a ser pesquisado deve estar nela da mesma maneira que foi armazenado, caso não esteja, o relatório não irá exibi-lo. Por exemplo, é necessário trazer os dados da cliente Vanessa; se esse nome for cadastrado de seguinte maneira – VaNeSsA – e procurarmos por Vanessa, o banco irá informar que a cliente não existe. E agora? Com a aplicação de uma função caracter podemos realizar a tarefa sem problemas:

1. `SELECT nome FROM funcionario WHERE UPPER(nome) = 'VANESSA';`

Assim, os dados são transformados em maiúsculos e pesquisados da forma que indicamos, em maiúsculo também.

Função SUBSTR – retorna parte de uma string.

Sintaxe: `SUBSTR(string, m [, n])`

Onde:

String: palavra ou coluna do tipo caracter que deve ser cortada.

M: especifica a posição inicial.

N: especifica quantos caracteres serão cortados a partir da posição inicial.

Exemplo:

1. `SELECT nome, substr(nome,2,4) FROM funcionario;`

Função LENGTH – retorna o comprimento, em caracteres, de um string.

Sintaxe: `LENGTH(nome da coluna)`

Exemplo:

1. `SELECT nome, LENGTH (nome) "Tamanho" FROM funcionario;`

Referências

BEIGHLEY, Lynn. *Use a cabeça SQL*. Rio de Janeiro: AltaBooks.

FANDERUFF, Damaris. *Dominando o Oracle 9i: modelagem e desenvolvimento*. São Paulo: Makron.

GRAVES, Mark. *Projeto de banco de dados com XML*. 1. ed. São Paulo: Pearson, 2003.

MORELLI, Eduardo Terra. *Oracle 9i fundamental: SQL, PL/SQL e administração*, São Paulo: Érica.

PRICE, Jason. *Oracle Database 11g SQL/Jason Price*. Tradução: João Eduardo Nóbrega Tortello. Porto Alegre: Bookman, 2009.

SILVA, Robson S. *Oracle Database 10g Express Edition*. 1. ed. São Paulo: Érica, 2007.



Avalie este tópico



ANTERIOR

Criação de relatórios utilizando filtros, operadores aritméticos e de banco de dados

Biblioteca

([https://www.uninove.br/conheca-](https://www.uninove.br/conheca-a-uninove/biblioteca/sobre-a-biblioteca/apresentacao/)

a-

uninove/biblioteca/sobre-

a-

biblioteca/apresentacao/)

Portal Uninove

(<http://www.uninove.br>)

Mapa do Site



Índice

Criação de relatórios utilizando filtros, operadores e funções em banco de dados

© Todos os direitos reservados

Ajuda?

PRÓXIMO
(<https://ava.uninove.br/cursos/>)

