Deusdeth Mariano
E-mail: deusdeth.mariano@ceub.edu.br

UniCEUB
Centro Universitário de Brasília

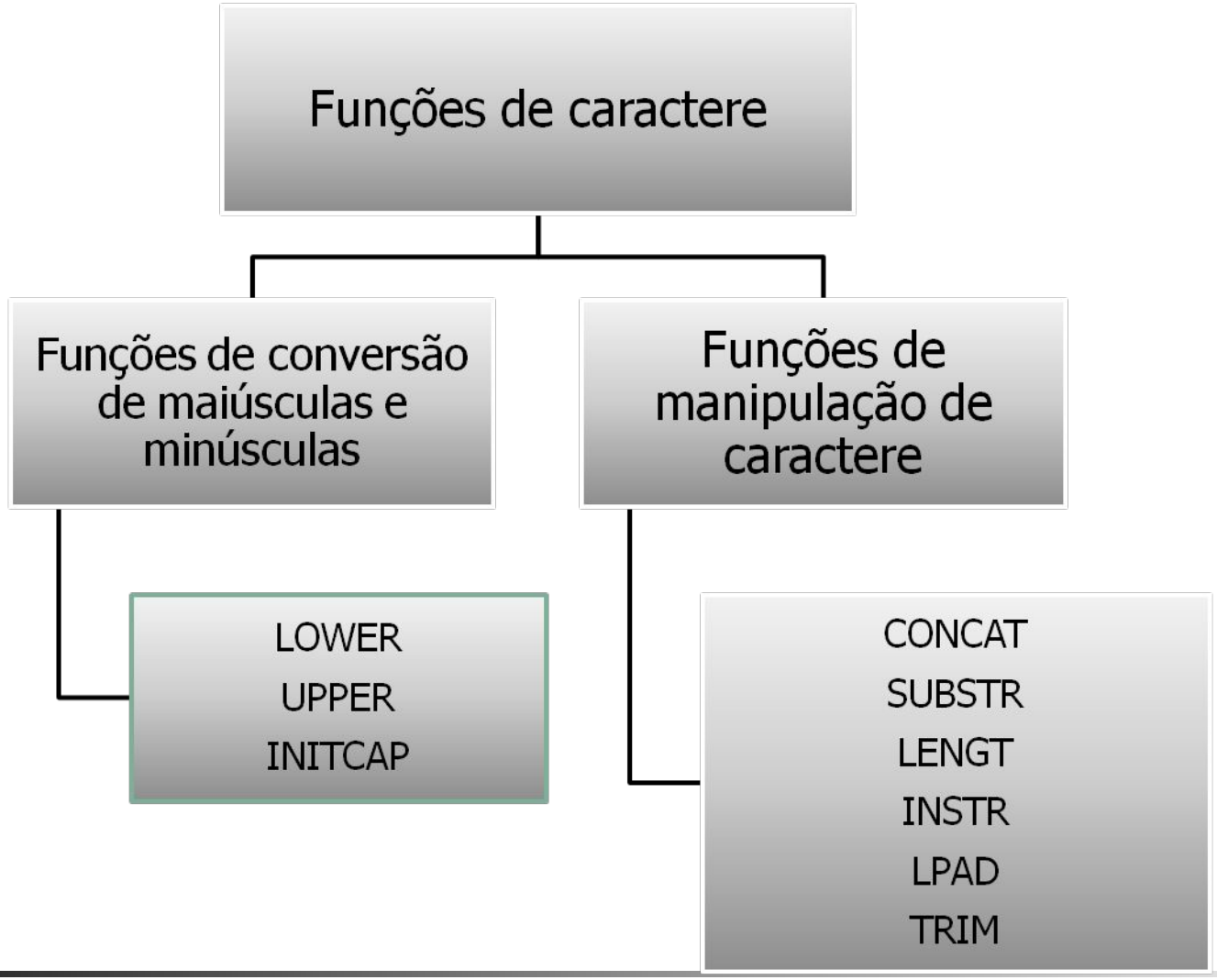
SQL - DML - Funções

Funções de uma única linha

- Manipulam itens de dados.
- Aceitam argumentos e retornam um valor.
- Agem em cada linha retornada.
- Retornam um resultado por linha.
- Podem modificar o tipo de dados.
- Podem ser aninhadas.

```
function_name (coluna|expressão, [arg1, arg2, ...])
```

Funções de caractere



Objetivos das funções

Função	Objetivo
LOWER(coluna|expressão)	converte valores alfanuméricos para letras minúsculas.
UPPER(coluna|expressão)	converte caractere alfanuméricos para letras maiúsculas.
INITCAP(coluna|expressão)	converte a primeira letra de cada palavra para maiúscula e mantém as outras letras minúscula.
CONCAT(coluna|expressão)	Concatena o primeiro valor do caractere ao segundo valor do caractere, equivalente ao operador de concatenação(||).

Objetivos das funções

Função	Objetivo
SUBSTR(coluna|expressão,m[,n])	Extrai uma string de determinado tamanho na expressão começando na posição m, até n caracteres depois (n pode ser omitido, aí se pega todos os caracteres até o final da string.
LENGTH(coluna|expressão)	Retorna o número de caracteres da expressão
INSTR(coluna|expressão,m)	Retorna a posição numérica do caractere nomeado.
LPAD(coluna|expressão,n,'string')	Preenche o campo a esquerda com o valor informado com uma largura total de n posições.
TRIM(anterior|posterior|ambos, trim_character FROM trim_source)	organiza os caracteres de fim de linha, a partir de uma string de caracteres.

Exemplo de manipulação de caractere

Função	Resultado
LOWER('Curso de SQL Oracle')	curso de sql oracle
UPPER('Curso de SQL Oracle')	CURSO DE SQL ORACLE
INITCAP('Curso de SQL Oracle')	Curso De Sql Oracle
CONCAT('bom',' dia')	bom dia
SUBSTR('exemplo',1,4)	exem
LENGTH('exemplo')	7
INSTR('exemplo','p')	5
LPAD(sal,10,'*')	******5000
TRIM('s' FROM 'ssmith')	mith

A tabela Dual

- Uma tabela "fictícia" utilizada no Oracle para retornar valores de resultados conhecidos.
  - Por ex., valores de constantes, expressões não derivadas de uma tabela com dados do usuário.
- A tabela DUAL pertence ao usuário "SYS" e pode ser acessada por todos os usuários.
  - Ela contém uma coluna, DUMMY, e uma linha com valor X que será populado com o valor de retorno de interesse na seleção.
- Exemplo, selecione a data e hora do servidor:

```
SELECT
   TO_CHAR(SYSDATE, 'DD/MON/YYYY HH24:MI:SS')
FROM DUAL;
```

Exemplos de manipulação de caractere (cont.)

```
SELECT LOWER('Curso de SQL Oracle')
   "Colocando em Minúsculas" FROM
  DUAL;

SELECT UPPER('Curso de SQL Oracle')
   "Colocando em Maiúsculas" FROM
  DUAL;

SELECT SUBSTR('exemplo',1,4)FROM DUAL;

SELECT LENGTH('exemplo') FROM DUAL;
```

Exemplos de manipulação de caractere (cont.)

```
SELECT 'O cargo de ' ||
  INITCAP(nomeemp) ||
      ' é de ' ||
      LOWER(cargo)
      AS "Detalhes do Empregado"
FROM empregado;

SELECT matricula, nomeemp, coddepto
FROM empregado
WHERE nomeemp = 'martins'
```

Exemplos de manipulação de caractere (cont.)

```
SELECT matricula, nomeemp, coddepto
FROM empregado
WHERE nomeemp = UPPER('martins');

SELECT nomeemp "Nomes que terminam
  com N"
FROM empregado
WHERE SUBSTR(nomeemp, -1, 1) = 'N';
```

Funções numéricas

- ROUND(*coluna|expressão,n*): arredonda a coluna, expressão ou valor para *n* casas decimais, ou se n for omitido, nenhuma casa decimal.
  - Se n for negativo, os números à esquerda do ponto decimal serão arredondados.

ROUND(45.926, 2) → 45,93

```
SELECT ROUND(45.926, 2)
   "Valor Arredondado" FROM DUAL;
```

Funções numéricas

- TRUNC(*coluna|expressão,n*): trunca a coluna, expressão ou valor para *n* casas decimais ou se n for omitido, nenhuma casa decimal.
  - Se n for negativo, os números à esquerda do ponto decimal serão truncados.

TRUNC (45.926, 2) → 45,92

```
SELECT TRUNC (45.926, 2)
   "Valor Truncado" FROM DUAL;
```

- MOD(*m,n*): retorna o resto da divisão de *m* por *n*.

MOD (1600, 300) → 100

```
SELECT MOD(1600,300) "Resto de 1600/300"
FROM DUAL;
```

Trabalhando com Datas

- O valores de data são armazenados no tipo date. O tipo date armazena data contendo dados como século, ano, mês, dia, hora, minutos e segundo.
  - Podem representar valores entre 1 de Janeiro de 4712 AC e 31 de Dezembro de 4712 DC.
- A data é guardada internamente através de um formato de dados desconhecido para o utilizador/programador.
  - No entanto, é possível alterar o formato de apresentação conforme time zones (fusos horários).
  - O formato padrão para a instalação Oracle no território BRAZIL é DD/MM/RR (Dia/Mês/Ano em 2 ou 4 posições).

Aritmética com Datas

Podem ser executadas as seguintes operações:

Operação	Resultado	Descrição
data + número	Data	Adiciona um número de dias para uma data
data – número	Data	Subtrai um número de dias de uma data
data – data	Número de dias	Subtrai uma data de outra
data + número/24	Data	Adiciona um número de hora para uma data

```
SELECT nomeemp, (SYSDATE-dataadm)/7 semanas
FROM empregado
WHERE coddepto = 10;
```

Funções de Data

Função	Descrição
MONTHS_BETWEEN(data1,data2)	Retorna o número de meses entre data1 e data2.
ADD_MONTHS(data,n)	Adiciona um número n de meses de calendário à data.
NEXT_DAY(data,'dia_semana')	Retorna o valor de data do próximo dia da semana especificado por dia_semana ('segunda', 'terça',...). Também se podem usar os números de 1 (Domingo) a 7(Sábado).
LAST_DAY(data)	Localiza a data do último dia do mês que contém a data.
SYSDATE	Função sem argumentos que devolve a data e hora do servidor.

Funções de Data (cont.)

Função	Descrição
ROUND(data, [‘DAY’|‘MONTH’|‘YEAR’|‘outro formato*’])	Retorna a data arredondada para a unidade de formato especificada no segundo argumento. Arredonda a data para o dia ou ao dia da semana ou ao mês ou ao ano.
TRUNC(data, [‘DAY’|‘MONTH’|‘YEAR’|‘outro formato*’])	Retorna a data truncada para a unidade de formato especificada no segundo argumento. Trunca a data para o dia ou ao dia da semana ou ao mês ou ao ano.

* Outros formatos de datas serão apresentados mais adiante.

Usando funções de data

```
SELECT MONTHS_BETWEEN('31/12/09', SYSDATE)
   "Meses para o final do ano" FROM DUAL;

SELECT ADD_MONTHS(SYSDATE, -3)
   "Três meses atrás" FROM DUAL;

SELECT NEXT_DAY(SYSDATE,'SEXTA')
   "Próxima Sexta-Feira" FROM DUAL;

SELECT LAST_DAY(SYSDATE)
   "Ultimo dia deste mês" FROM DUAL;
```

Usando funções de data (cont.)

```
SELECT TRUNC(SYSDATE, 'MM')
    "Início do mês" FROM DUAL;

SELECT
  TRUNC(TO_DATE('15/05/2008'),'YEAR')
   "Início do ano" FROM DUAL;

SELECT ROUND(TO_DATE('15/05/2008'),
  'MONTH') "Início do mês" FROM DUAL;
```

Usando funções de data (cont.)

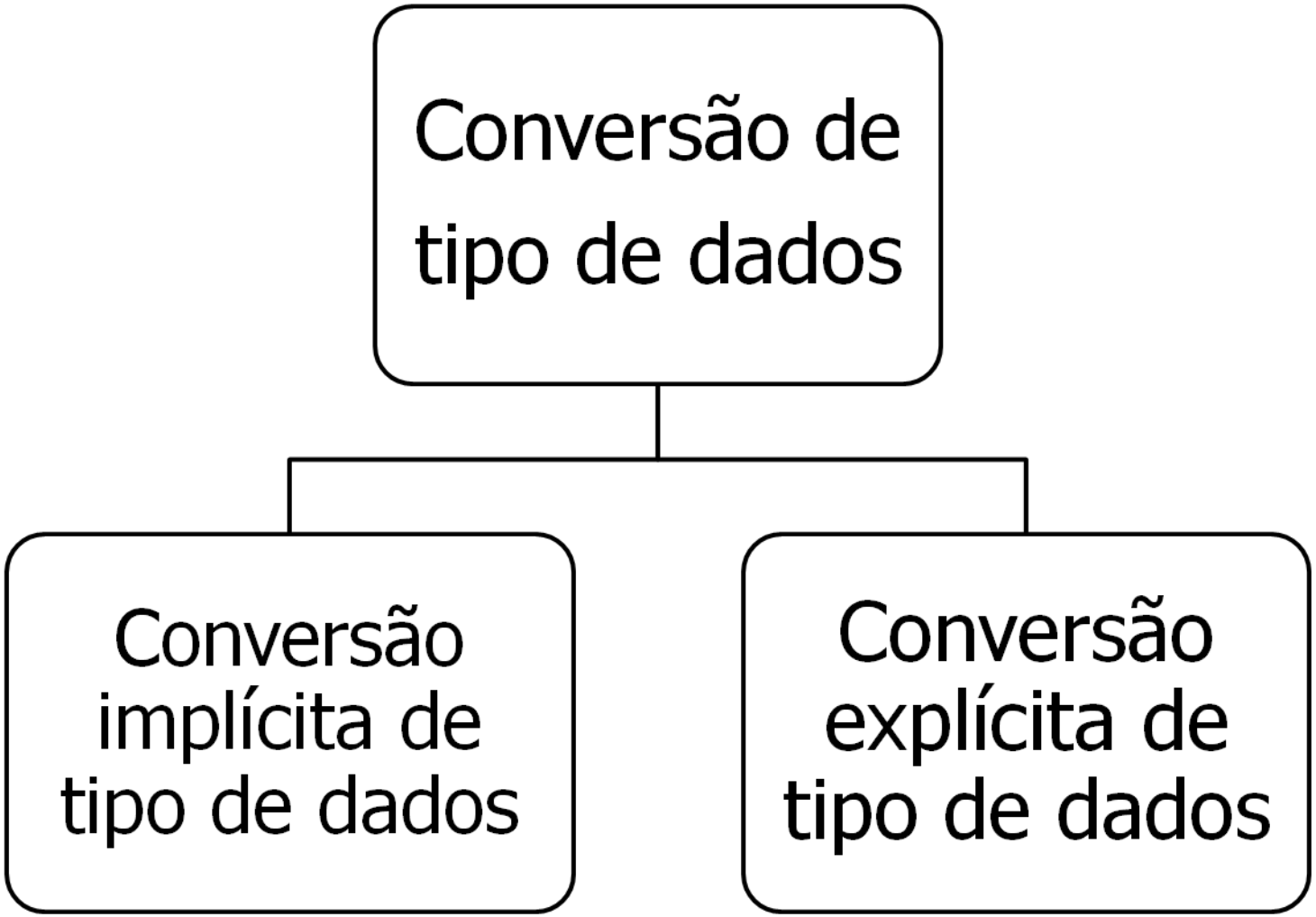
- Recupere as datas dos próximos dias da semana.

```
SELECT  NEXT_DAY(sysdate,'Domingo')
  "Domingo",
        NEXT_DAY(sysdate,'Segunda')
  "Segunda",
        NEXT_DAY(sysdate,'Terça') "Terça",
        NEXT_DAY(sysdate,'Quarta')"Quarta",
        NEXT_DAY(sysdate,'Quinta')"Quinta",
        NEXT_DAY(sysdate,'Sexta') "Sexta",
        NEXT_DAY(sysdate,'Sábado')"Sábado"
FROM DUAL;
```

Usando funções de data (cont.)

- Exemplo:
  - Compare as datas de admissão de todos os funcionários contratados em 2001. Exiba o número do funcionário, a data de admissão e o mês de início usando as funções ROUND e TRUNC.

Funções de conversão



Conversão implícita de tipos de dados (cont)

De	Para
VARCHAR2 ou CHAR	NUMBER
VARCHAR2 ou CHAR	DATE
NUMBER	VARCHAR2
DATE	VARCHAR2

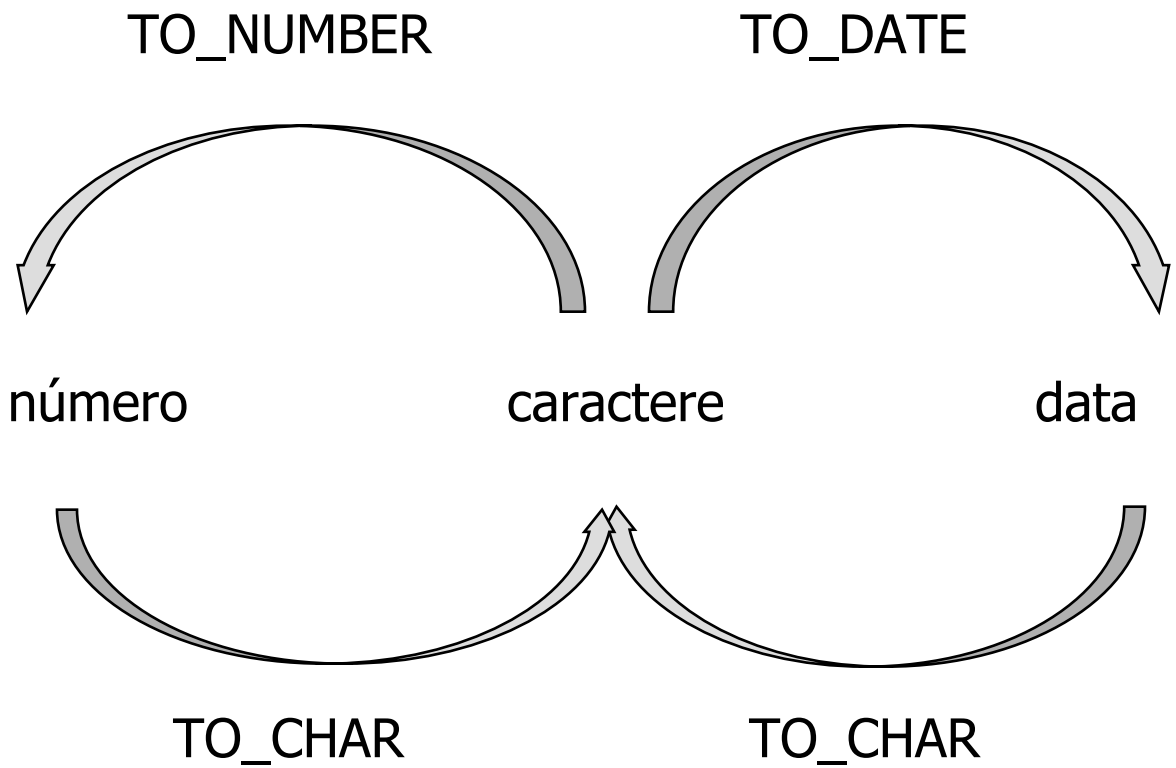
Conversão implícita de tipos de dados

- Para avaliação da expressão, o servidor Oracle pode converter automaticamente o seguinte:

De	Para
VARCHAR2 ou CHAR	NUMBER
VARCHAR2 ou CHAR	DATE

Conversões de CHAR para NUMBER ocorrem somente se a string de caractere representar um número válido. Conversões de CHAR para DATE ocorrem somente se a string de caractere possuir o formato default da instalação Oracle (no nosso caso, DD/MM/RR.)

Conversão explícita de tipos de dados (cont)



Conversão explícita de tipos de dados (cont)

Função	Objetivo
TO_CHAR(número|data, [fmt],[nlsparams])	Converte um valor de número ou data para uma string de caractere VARCHAR2 com modelo de formato fmt (que será descrito mais adiante) No caso de um número, o parâmetro nlsparams especifica o caractere decimal, separador de grupo, símbolo de moeda. No caso de uma data, o parâmetro nlsparams especifica o idioma no qual os nomes de dias, meses e abreviação retornam.
TO_NUMBER(carac,[fmt], [nlsparams])	Converte uma string de caractere contendo dígitos para um número no formato especificado pelo modelo de formato opcional fmt. O parâmetro nlsparams tem a mesma finalidade de TO_CHAR para números.

Conversão explícita de tipos de dados (cont)

Função	Objetivo
TO_DATE(carac,[fmt], [nlsparams])	Converte uma string de caractere representando uma data para um valor de data de acordo com o fmt especificado. O parâmetro nlsparams possui a mesma finalidade na função TO_CHAR para datas.

Função TO_CHAR com datas

TO_CHAR (data, '*fmt*')

- O modelo de formato:
  - Deve estar entre aspas e fazer distinção entre maiúsculas e minúsculas.
  - Pode incluir qualquer elemento de formato de data válido.
  - Tem um elemento *fm* para remover espaços preenchidos ou suprimir zeros à esquerda.
  - É separado do valor de data por uma vírgula.

Elementos de formatos de data

Elemento	Descrição
DDD ou DD ou D	Dia do ano, mês ou semana
DAY	Dia da semana (extenso)
MM	Número do mês
MON	Nome abreviado do mês
MONTH	Nome por extenso do mês
YYYY, YYY, YY, Y	Representar o ano com 4, 3, 2 ou 1 dígitos respectivamente
HH ou HH12 ou HH24	Horas do dia ou hora (1 a 12) ou hora (0 a 23)
MI	Minutos
SS	Segundos
SSSSS	Segundos depois da meia-noite
- / , . ; : "texto”	Pontuação ou texto entre aspas

Exemplo de uso do *fmt* com datas

```
SELECT nomeemp,
   TO_CHAR(dataadm,
      'fmDD "de" MONTH "de" YYYY')
       "Data de Admissão"
FROM empregado;

SELECT TO_CHAR(SYSDATE, 'DD/MM/YY "às"
  HH24:MM:SS"hs"') "Hora e Data do
  Sistema"  FROM DUAL;

SELECT TO_CHAR(SYSDATE, 'Day, DD "de"
  Month "de YYYY') "Data com Dia da
  Semana"  FROM DUAL;
```

Função TO_CHAR com números

TO_CHAR (número, 'fmt')

- Use estes formatos com a função TO_CHAR para exibir um valor de número como um caractere:

Elemento	Descrição	Exemplo	Resultado
9	Um símbolo 9 para cada algarismo significativo a representar antes ou depois da vírgula	999999	1234
0	Exibe zeros à esquerda	0999999	001234
$	Sinal de dólar flutuante	$999999	$1234
L	Símbolo da moeda local flutuante	L999999	FF1234
.	Ponto decimal na posição especificada	999999.99	1234.00
,	Vírgula na posição especificada	999,999	1,234

Usando a função TO_CHAR com números

```
SELECT TO_CHAR(sal,'$999,999')
  Salario
FROM empregado
WHERE nomeemp = 'BARROS';


SELECT
  TO_CHAR(-1000000,'L99999,99MI')
  "Valor Negativo na Moeda Local”
FROM DUAL;
```

Usando as funções TO_NUMBER e TO_DATE

```
SELECT TO_NUMBER('1210.73', '9999.99')
FROM DUAL;

SELECT nomeemp "Contratados no Mês de
  Dezembro"
FROM empregado
WHERE TO_NUMBER(TO_CHAR(dataadm,'MM'))=
  12;

SELECT TO_DATE('24.06.2009', 'DD/MM/YY')
  "Data"
FROM DUAL;
```

Função NVL

- Converte nulo para um valor real.
- Sintaxe:

```
NVL(expr1, expr2)
```

Onde,

expr1 é o valor de origem ou expressão que pode ser nulo.

expr2 é o valor de destino para a conversão de nulo.

- Os tipos de dados que podem ser usados são data, caractere e número.

Tipo de Dados	**Exemplo de Conversão**
NUMBER	NVL(comm,0)
DATE	NVL(dataadm,'01-JAN-97')
CHAR ou VARCHAR2	NVL(cargo,'Sem cargo ainda')

Função DECODE

- Facilita pesquisas condicionais realizando o trabalho de uma instrução case ou if-then-else.

```
DECODE (col/exp, pesquisa1, resultado1
                [, pesquisa2,
  resultado2,...,]
                [, default])
```

  - O primeiro parâmetro é comparado com pesquisa1. Se for igual é devolvido o valor de resultado1. Caso contrário é comparado com pesquisa2 e assim sucessivamente. Se não for igual a nenhum dos os valores de pesquisa então é devolvido o valor default.

Usando a função NVL

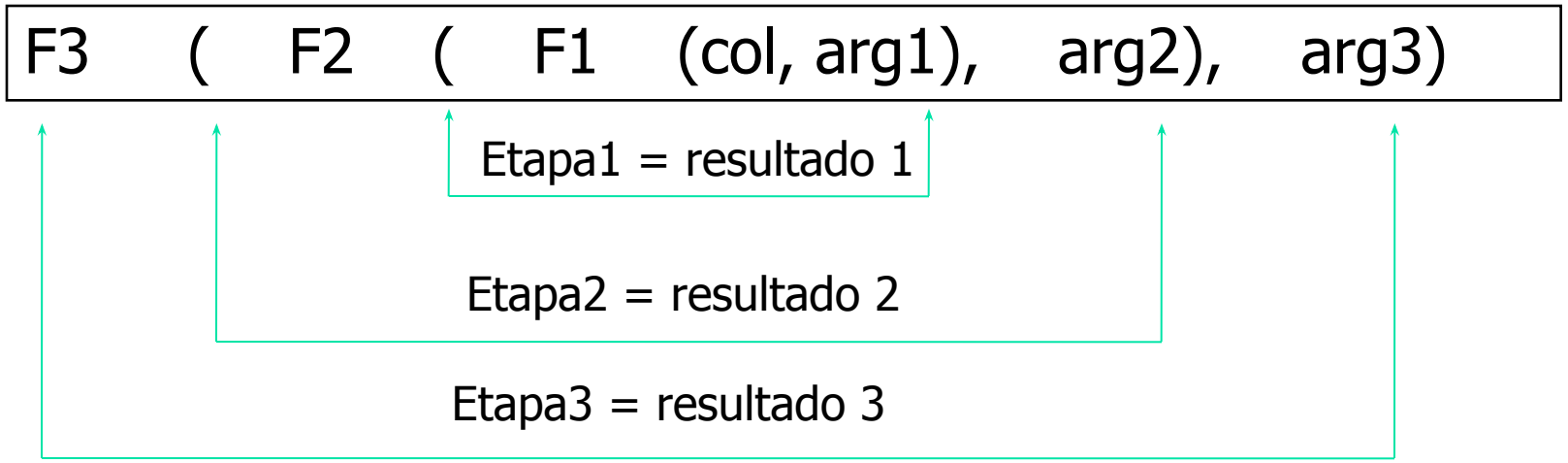
```
SELECT
     nomeemp,
     sal,
     comm,
     (sal*12)+ NVL(comm,0)
FROM empregado;
```

Usando a função DECODE

```
SELECT nomeemp, cargo, sal,
  DECODE (cargo, 'ANALISTA',
  sal*1.1,
         'CAIXA',   sal*1.15,
         'GERENTE',   sal*1.2,
        sal)
 "Novo Salário"
FROM empregado;
```

Aninhando funções

- As funções de uma única linha podem ser aninhadas em qualquer nível. Funções aninhadas são avaliadas a partir do nível mais interno para o nível mais externo.




```
SELECT nomeemp, NVL(TO_CHAR(gerente),
   'Sem gerente')
FROM empregado WHERE gerente IS NULL;
```

Exercícios

1. Crie uma consulta para exibir a data atual. Coloque um apelido na coluna Date.
2. Exiba o número do funcionário, o nome, o salário e o aumento salarial de 15% expresso com número inteiro. Coloque um apelido na coluna como Novo Salario. Salve a instrução em um arquivo nomeado slide38.sql. Execute a consulta.
3. Modifique a consulta do arquivo slide38.sql para adicionar uma coluna que subtrairá o salário antigo do novo.Coloque um apelido nesta nova coluna denominado aumento.
4. Para cada funcionário exiba o nome do mesmo e calcule o número de meses entre hoje e as sua datas de admissão. Coloque um apelido na coluna denominado MesesTrabalhados. Ordene os resultados por número de meses trabalhados. Arredonde para cima o número de meses para o número inteiro mais próximo.
5. Crie uma consulta que exibirá o nome do funcionário com a primeira letra maiúscula e todas as outras minúsculas, bem como o tamanho de seus nomes. Faça isto para todos os funcionários cujo nome começa com J, A ou M. Forneça a cada coluna um rótulo apropriado.

Funções de grupo

- As funções de grupo operam em conjuntos de linhas para fornecer um resultado por grupo.
  - Esses conjuntos podem ser a tabela inteira ou a tabela dividida em grupos.
- Sintaxe

```
 SELECT [DISTINCT]
group_function(column1), column2,
group_function(column3)...
  FROM table_list
  [WHERE conditions]
  [GROUP BY group_by_list]
  [ORDER BY order_list [ASC | DESC] ]
```

Funções de grupo (cont.)

- Onde,
  - group_function: é uma das seguintes funções de agregação: AVG, COUNT, MAX, SUM, etc.
  - GROUP BY group_by_list: especifica colunas cujos valores determinam a base para formação de grupos em linhas. As linhas em cada grupo tem um único valor em group_by_list. A lista de grupos de group_by_list pode ter uma ou mais colunas ou expressões.

Diretrizes para funções de grupo

- Todas as funções de grupo, exceto COUNT(*), ignoram valores nulos. Para substituir um valor por valores nulos, use a função NVL.
- Ao usar uma cláusula WHERE, você pode excluir linhas com antecedência antes de dividi-las em grupos.

Diretrizes para funções de grupo

- Ao usar a cláusula GROUP BY:
  - Se você incluir uma função de grupo em uma cláusula SELECT, não poderá selecionar resultados individuais, a menos que a coluna individual apareça na cláusula GROUP BY. Se você não conseguir incluir a lista de colunas, uma mensagem de erro será exibida.
  - Você deve incluir as colunas na cláusula GROUP BY.
  - Não é possível usar o apelido de coluna na cláusula GROUP BY.
  - Por default, as linhas são classificadas por ordem crescente das colunas incluídas na lista GROUP BY. Isso pode ser sobreposto usando a cláusula ORDER BY.

Funções de grupo

Função	Descrição
AVG([DISTINCT|ALL]n)	Valor médio de n, ignorando valores nulos
COUNT({*|[DISTINCT|ALL]expr})	Número de linhas, onde expr avalia para algo diferente de nulo (Conte todas as linhas selecionadas usando *, inclusive duplicadas e linhas com nulos.)
MAX([DISTINCT|ALL]expr)	Valor máximo de expr, ignorando valores nulos
MIN([DISTINCT|ALL]expr)	Valor mínimo de expr, ignorando valores nulos
SUM([DISTINCT|ALL]n)	Valores somados de n, ignorando valores nulos

Exemplos de uso de funções agregadas

```
SELECT AVG(sal), MAX(sal),
  MIN(sal), SUM(sal)
FROM empregado
WHERE cargo LIKE 'VEND%';

SELECT MIN(dataadm), MAX(dataadm)
FROM empregado;
```

Exemplos de uso de funções agregadas

```
SELECT COUNT(*) as "Número de
  Linhas"
FROM empregado
WHERE coddepto = 30;

SELECT COUNT(comm) as "Número de
  Linhas não Nulas"
FROM empregado
WHERE coddepto = 30;
```

Usando a cláusula GROUP BY

- Todas as colunas na lista SELECT que não estejam em funções de grupo devem estar na cláusula GROUP BY.

```
SELECT coddepto, AVG(sal)
FROM empregado
GROUP BY coddepto;
```

  - A cláusula GROUP BY especifica como as linhas devem ser agrupadas. As linhas são agrupadas pelo número do departamento, de forma que a função AVG que esteja sendo aplicada à coluna de salários *calcule o salário médio para cada departamento.*

Usando a cláusula GROUP BY (Cont.)

- Sem os números dos departamentos, entretanto, os resultados não parecem significativos. Você pode usar a função de grupo na cláusula ORDER BY.

```
SELECT coddepto, AVG(sal)
FROM empregado
GROUP BY coddepto
ORDER BY AVG(sal);
```

Usando a cláusula GROUP BY (Cont.)

- A coluna GROUP BY não precisa estar na lista SELECT. Por exemplo, a instrução SELECT abaixo exibe os salários médios para cada departamento sem exibir os respectivos números dos departamentos.

```
SELECT AVG(sal)
FROM empregado
GROUP BY coddepto;
```

Usando a cláusula GROUP BY em várias colunas

- É possível retornar resultados sumários para grupos e subgrupos listando mais de uma coluna GROUP BY.

```
SELECT coddepto, cargo, sum(sal)
FROM empregado
GROUP BY coddepto, cargo;
```

- A cláusula GROUP BY especifica como você deve agrupar as linhas. No exemplo:
  - Primeiro, as linhas são agrupadas pelo número do departamento.
  - Em seguida, dentro dos grupos de números de departamentos, as linhas são agrupadas pelo cargo.

Exercícios

1. Exiba os salários maior, médio, menor e a soma de todos os salários de todos os funcionários. Coloque um rótulo nas colunas Máximo, Mínimo, Soma e Média, respectivamente. Arredonde os resultados para o número inteiro mais próximo. Salve a instrução SQL em um arquivo chamado slide50.sql. Execute a consulta
2. Modifique o arquivo slide50.sql para exibir o salário maior, médio, menor e a soma de todos os salários para cada tipo de cargo. Salve novamente o arquivo com o nome slide50-2.sql. Execute novamente a consulta.
3. Crie uma consulta para exibir o número de pessoas com o mesmo cargo.
4. Determine o número de gerentes sem listá-los. Coloque um rótulo na coluna - "Número de Gerentes
5. Crie uma consulta para exibir a diferença entre os maiores e menores salários. Coloque um rótulo na coluna "DIFERENÇA".
6. Crie uma consulta para exibir o nome do departamento, o nome da localização, o número de funcionários e o salário médio de todos os funcionários do departamento. Arredonde o salário médio para duas casas decimais apenas.

FIM