

Bases de Dados

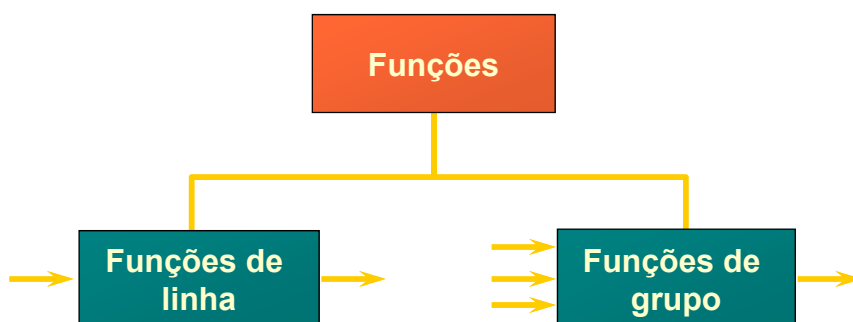
SQL- Funções de Linha



Funções do SQL



Dois Tipos de Funções do SQL



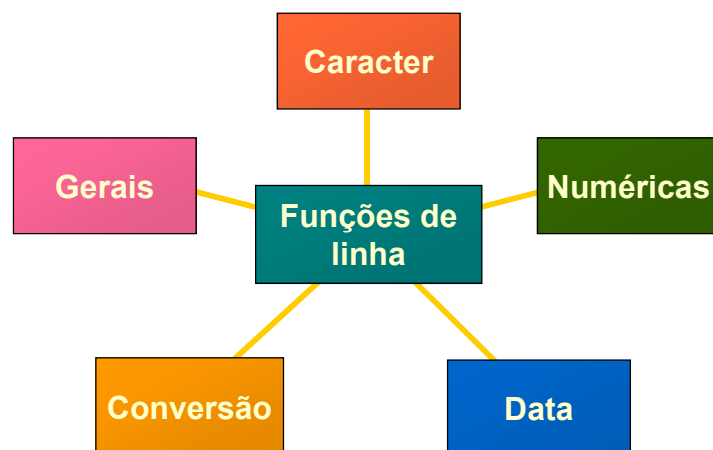
Funções de Linha

- ▶ Manipulam os items de dados
- ▶ Aceitam os argumentos e devolvem um valor
- ▶ Actuam em cada linha devolvida
- ▶ Devolvem um resultado por linha
- ▶ Podem modificar o tipo de dados
- ▶ Podem ser aninhadas
- ▶ Aceitam argumentos que podem ser uma coluna ou uma expressão

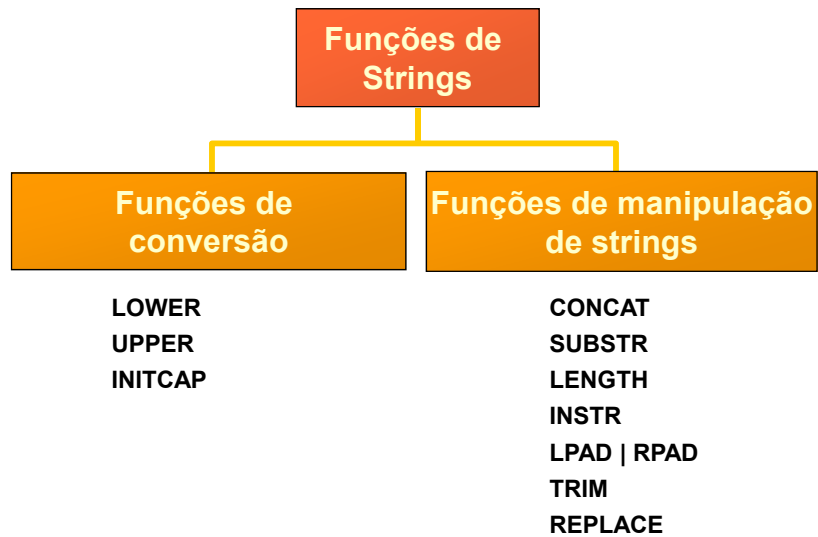
```
function_name (column|expression, [arg1, arg2,...])
```



Funções de Linha



Funções de Strings



▶ 7



Funções de Conversão

- ▶ Converte as strings de caracteres

Função	Resultado
LOWER(' SQL Curso ')	sql curso
UPPER(' SQL Curso ')	SQL CURSO
INITCAP(' SQL Curso ')	Sql Curso

▶ 8



Usar Funções de Conversão

- Mostrar o número do empregado, nome, e número do departamento para o empregado Blake.

```
SQL> SELECT  nemp, nome, ndep
2 FROM      emp
3 WHERE     nome = 'blake';
no rows selected
```

```
SQL> SELECT  nemp, nome, ndep
2 FROM      emp
3 WHERE     nome = UPPER('blake');
```

NEMP	NOME	NDEP
7698	BLAKE	30



Funções de Manipulação de Strings

- Manipulam strings de caracteres

Função	Resultado
CONCAT('Boa', 'String')	BoaString
SUBSTR('String',1,3)	Str
LENGTH('String')	6
INSTR('String', 'r')	3
LPAD(sal,10, '*')	*****5000
TRIM('S' FROM 'SSMITH')	MITH



Usar as Funções de Manipulação de Strings

```
SQL> SELECT nome, CONCAT (nome, funcao), LENGTH(nome),
2      INSTR(nome, 'A')
3 FROM emp
4 WHERE SUBSTR(funcao,1,8) = 'VENDEDOR';
```

NOME	CONCAT (NOME, FUNCAO)	LENGTH (NOME)	INSTR (NOME, 'A')
MARTIN	MARTINVENDEDOR	6	2
ALLEN	ALLENVENDEDOR	5	1
TURNER	TURNERVENDEDOR	6	0
WARD	WARDVENDEDOR	4	2



Funções Numéricas

- ▶ ROUND: arredonda o valor para o decimal especificado

ROUND(45.926, 2) → 45.93

- ▶ TRUNC: Trunca o valor para o decimal especificado

TRUNC(45.926, 2) → 45.92

- ▶ MOD: Devolve o resto da divisão

MOD(1600, 300) → 100



Usar a Função ROUND

```
SQL> SELECT ROUND(45.923,2), ROUND(45.923,0),  
2         ROUND(45.923,-1)  
3 FROM DUAL;
```

ROUND(45.923,2)	ROUND(45.923,0)	ROUND(45.923,-1)	
45.92	46	50	



Usar a Função TRUNC

```
SQL> SELECT TRUNC(45.923,2), TRUNC(45.923),  
2         TRUNC(45.923,-1)  
3 FROM DUAL;
```

TRUNC(45.923,2)	TRUNC(45.923)	TRUNC(45.923,-1)	
45.92	45	40	



Usar a Função MOD

- ▶ Calcular o resto da razão do salário com os prémios para todos os empregados cuja função seja vendedor.

```
SQL> SELECT  nome, sal, premios, MOD(sal, premios)
2  FROM      emp
3  WHERE     funcao = 'VENDEDOR';
```

NOME	SAL	PREMIOS	MOD(SAL, PREMIOS)
MARTIN	1250	1400	1250
ALLEN	1600	300	100
TURNER	1500	0	1500
WARD	1250	500	250



Trabalhar com Datas

- ▶ Oracle guarda as datas num formato numérico interno: século, ano, mês, dia, horas, minutos, segundos.
- ▶ O formato de data default é DD-MON-YY.
- ▶ SYSDATE é uma função que devolve a data e a hora.
- ▶ DUAL é uma tabela fictícia usada para ver SYSDATE.



Aritmética com Datas

- ▶ Adicionar ou subtrair um número a ou a partir de uma data para um valor resultante *date*.
- ▶ Subtrair duas datas para encontrar o *número* de dias entre as duas datas.
- ▶ Adicionar *horas* a uma data, dividindo o número de horas por 24.



Usar Operadores Aritméticos com Datas

```
SQL> SELECT nome, (SYSDATE-data_entrada)/7 SEMANAS
2 FROM emp
3 WHERE ndep = 10;
```

NOME	SEMANAS
KING	830.93709
CLARK	853.93709
MILLER	821.36566



Funções de Data

Função	Descrição
MONTHS_BETWEEN	Número de meses entre duas datas
ADD_MONTHS	Adiciona os meses do calendário à data
NEXT_DAY	Próximo dia da data especificada
LAST_DAY	Último dia do mês
ROUND	Arredonda a data
TRUNC	Trunca a data



Usar Funções de Data

- **MONTHS_BETWEEN ('01-SEP-95','11-JAN-94')** → **19.6774194**
- **ADD_MONTHS ('11-JAN-94',6)** → **'11-JUL-94'**
- **NEXT_DAY ('01-SEP-95','FRIDAY')** → **'08-SEP-95'**
- **LAST_DAY('01-SEP-95')** → **'30-SEP-95'**

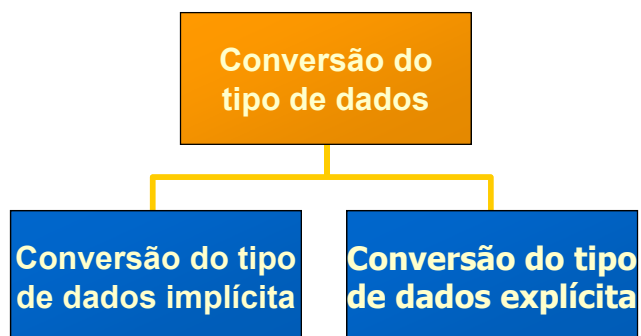


Usar Funções de Data

- **ROUND('25-JUL-95','MONTH') → 01-AUG-95**
- **ROUND('25-JUL-95','YEAR') → 01-JAN-96**
- **TRUNC('25-JUL-95','MONTH') → 01-JUL-95**
- **TRUNC('25-JUL-95','YEAR') → 01-JAN-95**



Funções de Conversão



Conversão do Tipo de Dados Implícita

- ▶ Nas atribuições o Oracle Server pode automaticamente converter o seguinte:

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



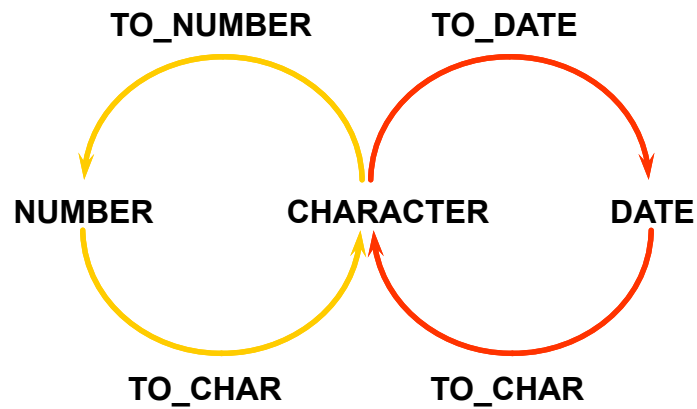
Conversão Implícita do Tipo de Dados

- ▶ No cálculo de expressões, o Oracle Server pode automaticamente converter o seguinte:

De	Para
VARCHAR2 ou CHAR	NUMBER
VARCHAR2 ou CHAR	DATE



Conversão do Tipo de dados Explícita



Função TO_CHAR com Datas

```
TO_CHAR(date, 'fmt')
```

- ▶ O modelo de formato:
 - ▶ Deve estar dentro de plicas e é case sensitive
 - ▶ Pode incluir qualquer elemento de formato de data válido
 - ▶ Tem um elemento *fm* para remover brancos ou zeros
 - ▶ Está separado do valor data por uma vírgula



Elementos do Modelo de Formato de Data

YYYY	Ano completo em números
YEAR	Ano soletrado
MM	Dois dígitos para o mês
MONTH	Nome completo do mês
DY	Abreviação com três letras do dia da semana
DAY	Nome completo do dia



Elementos do Modelo de formato de Datas

- Elementos da hora (Time) formatam a porção da hora da data.

HH24:MI:SS AM	15:45:32 PM
----------------------	--------------------

- Acrescentar strings de caracteres incluindo-as entre aspas.

DD "of" MONTH	12 of OCTOBER
----------------------	----------------------

- Sufixos do número soletram números.

ddspth	fourteenth
---------------	-------------------



Usar a Função TO_CHAR com datas

```
SQL> SELECT nome,  
2 TO_CHAR(data_entrada, 'fmDD Month YYYY') DATA_ENTRADA  
3 FROM emp;
```

NOME	DATA_ENTRADA
KING	17 November 1981
BLAKE	1 May 1981
CLARK	9 June 1981
JONES	2 April 1981
MARTIN	28 September 1981
ALLEN	20 February 1981
...	

14 rows selected.



Função TO_CHAR com Números

```
TO_CHAR(number, 'fmt')
```

- Usar estes formatos com a função TO_CHAR para mostrar o valor numérico como um caracter:

9	Representa um número
0	Força a mostrar um zero
\$	Coloca um sinal de dólar flutuante
L	Usa o símbolo de moeda local flutuante
.	Imprime um ponto decimal
,	Imprime um indicador de milhar



Usar a Função TO_CHAR com Números

```
SQL> SELECT TO_CHAR(sal, '$99,999') SALARIO
2 FROM emp
3 WHERE nome = 'SCOTT';
```

```
SALARIO
-----
$3,000
```



Funções TO_NUMBER e TO_DATE

- ▶ Converter uma string de caracteres num formato numérico usando a função **TO_NUMBER**

```
TO_NUMBER(char[, 'fmt'])
```

- ▶ Converter uma string de caracteres num formato data usando a função **TO_DATE**

```
TO_DATE(char[, 'fmt'])
```



Função NVL

► Converte null para um valor real

- Tipos de dados que não podem ser usados são date, character, e number.
- Os tipos de dados devem concordar
 - NVL(premios,0)
 - NVL(data_entrada,'01-JAN-97')
 - NVL(funcao,'Sem funcao')



Usar a Função NVL

```
SQL> SELECT nome, sal, premios, (sal*12)+NVL(premios,0)
2 FROM emp;
```

NOME	SAL	PREMIOS	(SAL*12)+NVL(PREMIOS,0)
KING	5000		60000
BLAKE	2850		34200
CLARK	2450		29400
JONES	2975		35700
MARTIN	1250	1400	16400
ALLEN	1600	300	19500
...			
14 rows selected.			



Função DECODE

- Facilita pesquisas condicionais fazendo o trabalho de uma instrução **CASE** ou **IF-THEN-ELSE**

```
DECODE(col/expression, search1, result1  
      [, search2, result2,...,]  
      [, default])
```



Usar a Função DECODE

```
SQL> SELECT funcao, sal,  
2      DECODE(funcao, 'ANALISTA', SAL*1.1,  
3              'CONTADOR',   SAL*1.15,  
4              'GESTOR',     SAL*1.20,  
5                          SAL)  
6      SALARIO_REVISTO  
7 FROM emp;
```

FUNCAO	SAL	SALARIO_REVISTO
PRESIDENTE	5000	5000
GESTOR	2850	3420
GESTOR	2450	2940
...		

14 rows selected.



Usar a Função DECODE

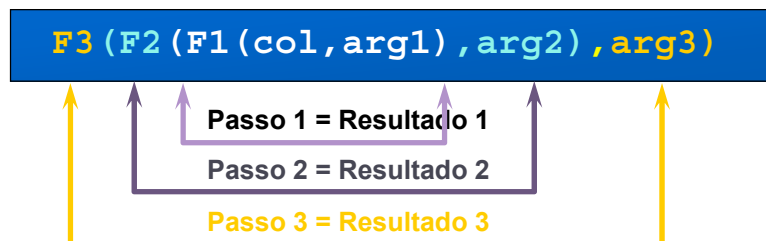
- Mostrar o valor da taxa aplicada a cada empregado do departamento 30.

```
SQL> SELECT nome, sal,
2          DECODE (TRUNC (sal/1000, 0),
3                  0, 0.00,
4                  1, 0.09,
5                  2, 0.20,
6                  3, 0.30,
7                  4, 0.40,
8                  5, 0.42,
9                  6, 0.44,
10                 0.45) VALOR_TAXA
11 FROM      emp
12 WHERE      ndep = 30;
```



Funções Aninhadas

- As funções de linha podem estar aninhadas até qualquer nível.
- As funções aninhadas são calculadas do nível mais interior para o mais exterior.



Funções Aninhadas

```
SQL> SELECT  nome,  
2          NVL(TO_CHAR(encar), 'Sem Encarregado')  
3 FROM      emp  
4 WHERE     encar IS NULL;
```

NOME	NVL(TO_CHAR(ENCAR), 'Sem encarregado')
KING	Sem Encarregado

