

Sistemas de Informação

Design e Desenvolvimento de Banco de Dados

PROF. MILTON

Funções de Uma Única Linha



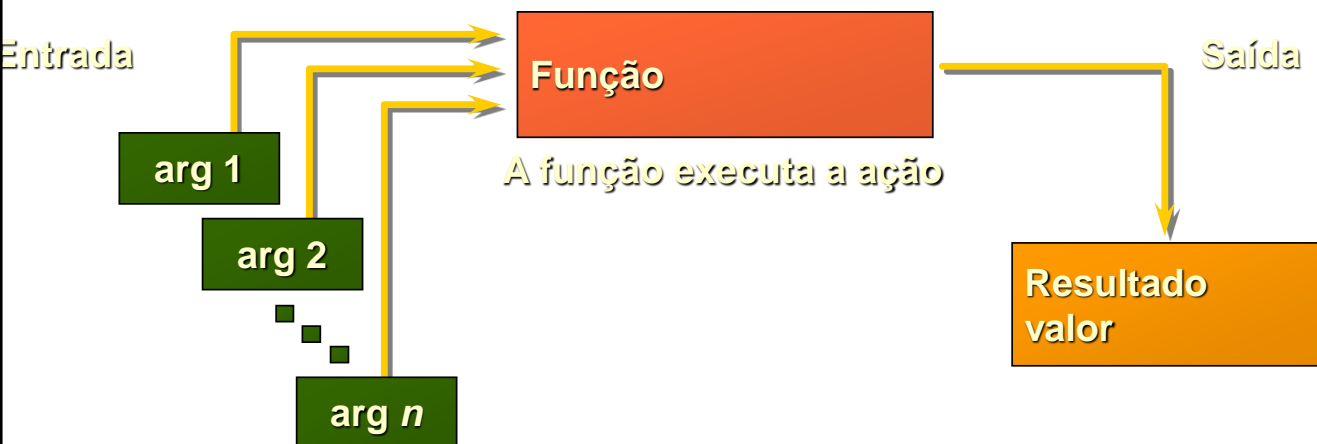
Objetivos

- Depois de completar esta lição, você poderá fazer o seguinte:
 - Descrever vários tipos de funções disponíveis no SQL
 - Usar funções de data, número e caractere nas instruções SELECT
 - Descrever o uso das funções de conversão

Objetivo da Lição

As funções formam o bloco de consulta básico mais avançado e são usadas para manipular valores de dados. Esta é a primeira de duas lições a explorar funções. Serão abordadas as funções de caractere de uma única linha, de número e de datas, bem como aquelas funções que convertem dados de um tipo para outro — por exemplo, dados de caractere para numérico.

Funções SQL



Funções SQL

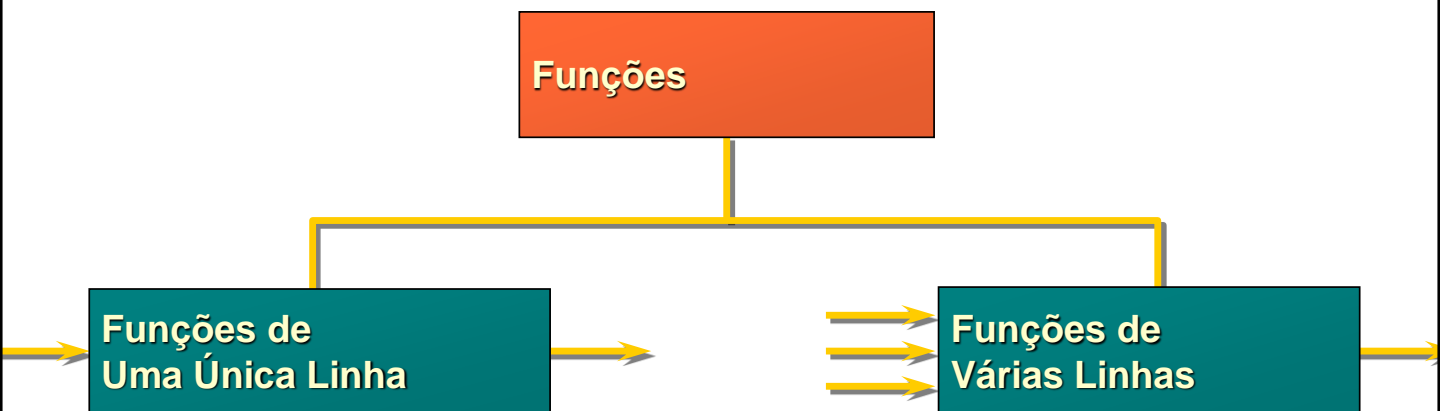
As funções são um recurso avançado do SQL e podem ser usadas para realizar o seguinte:

- Executar cálculos usando dados
- Modificar itens de dados individuais
- Manipular saída para grupos de linhas
- Formatar datas e números para exibição
- Converter tipos de dados de coluna

As **funções** SQL podem aceitar argumentos e sempre retornar um valor.

Observação: A maioria das funções descritas nesta lição são específicas para a versão SQL da Oracle.

Dois Tipos de Funções SQL



Funções SQL (continuação)

Há dois tipos distintos de funções:

Funções de Uma Única Linha

Funções de Várias Linhas

Funções de Uma Única Linha

Essas funções operam somente linhas únicas e retornam um resultado por linha. Há dois tipos diferentes de funções de uma única linha. Esta lição aborda as seguintes:

Caractere

Número

Data

Conversão

Funções de Várias Linhas

Essas funções manipulam grupos de linha a fim de obter um resultado por grupo de linhas.

Para obter mais informações, consulte o *Oracle Server SQL Reference*, para uma lista completa de funções e sintaxes disponíveis.

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 Uma Única Linha

Funções de uma única linha são usadas para manipular itens de dados. Elas aceitam um ou mais argumentos e retornam um valor para cada linha retornada pela consulta. Um argumento pode ser um dos seguintes:

- Constante fornecida pelo usuário
- Valor variável
- Nome da coluna
- Expressão

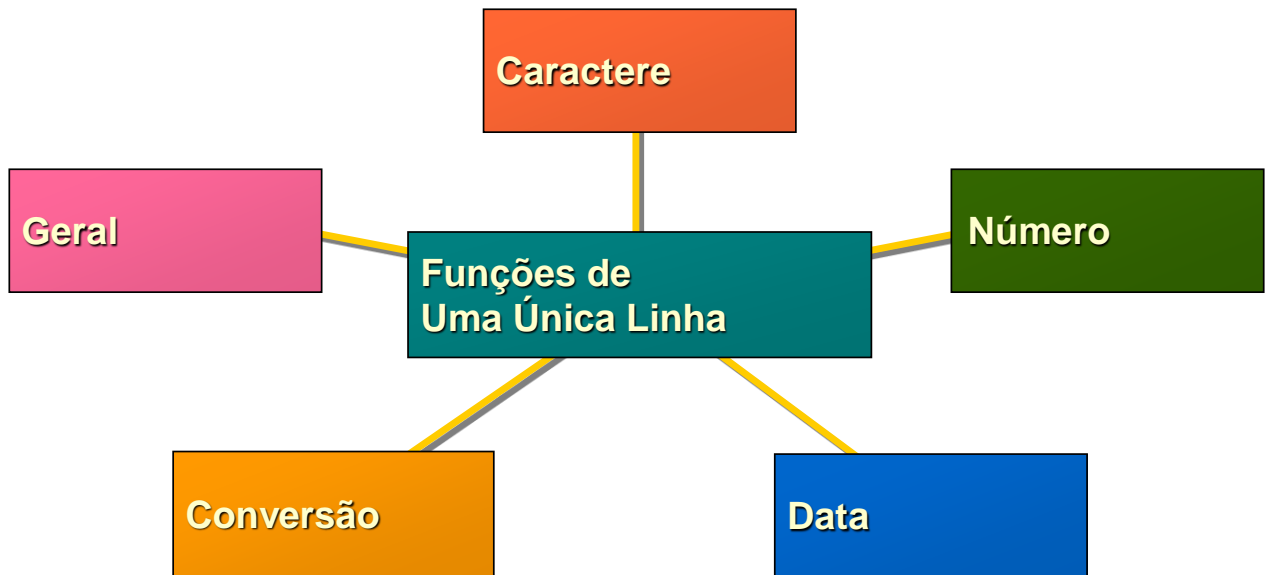
Os recursos de funções de uma única linha:

- Atuam em cada linha retornada na consulta
- Retornam um resultado por linha
- Podem retornar um valor de dados de um tipo diferente do mencionado
- Podem esperar um ou mais argumentos
- Podem ser usados em cláusulas SELECT, WHERE e ORDER BY; podem ser aninhados

Na sintaxe:

- function_name* é o nome da função
- coluna* é qualquer coluna de banco de dados nomeada
- expressão* é qualquer string de caractere ou expressão calculada
- arg1*, *arg2* é qualquer argumento a ser utilizado pela função

Funções de Uma Única Linha



Funções de Uma Única Linha (continuação)

Esta lição aborda as seguintes funções de uma única linha:

Funções de caractere: Aceitam entrada de caractere e podem retornar valores de número e caractere

Funções numéricas: Aceitam entrada numérica e retornam valores numéricos

Funções de data: Operam sobre valores de tipo de dados da data (Todas as funções de data retornam data um valor de tipo de dados de data exceto a função MONTHS_BETWEEN, que retorna um número.)

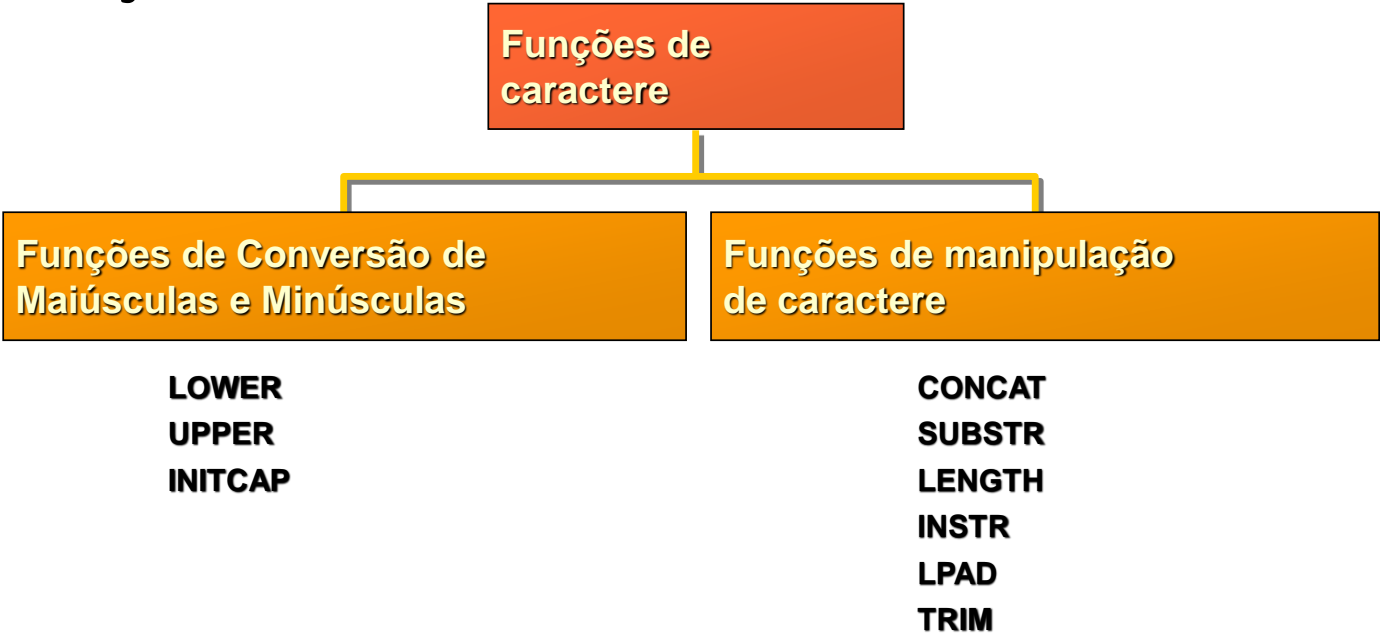
Funções de conversão: Convertem um valor de um tipo de dados para outro

Funções gerais:

Função NVL

Função DECODE

Funções de Caractere



Funções de Caractere

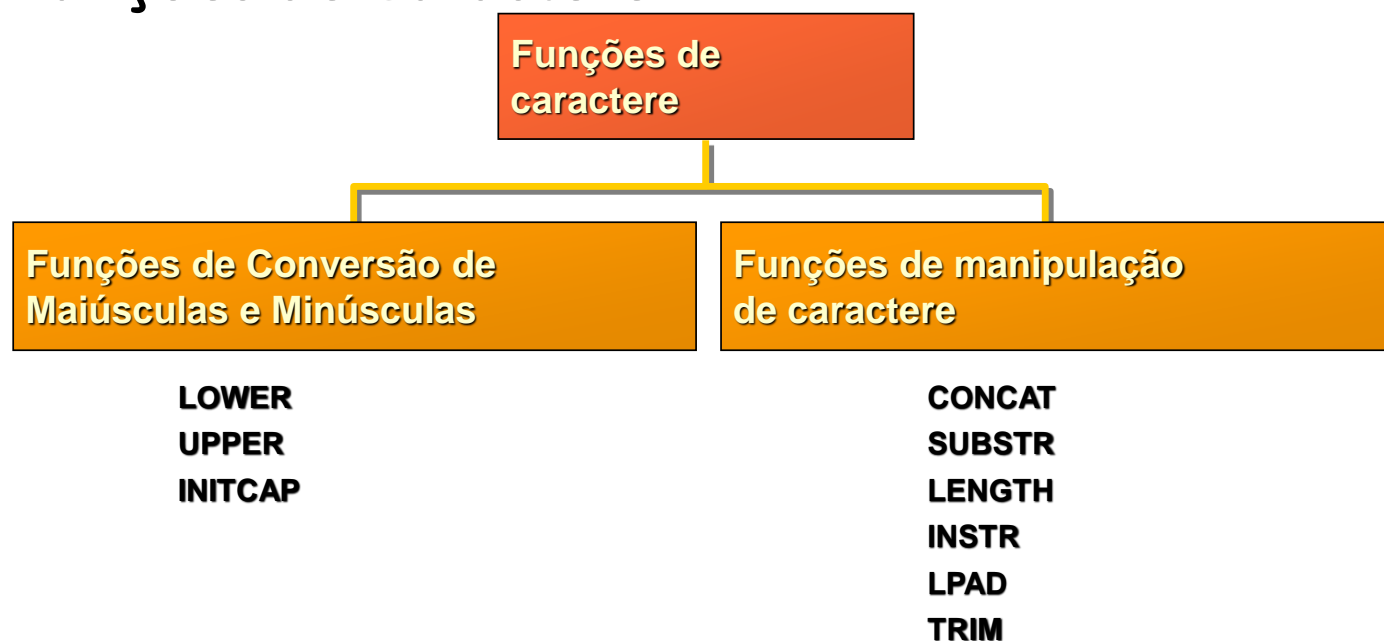
As **funções de caractere** de uma única linha aceitam dados de caractere como entrada e podem retornar valores de número e de caractere. As funções de caractere podem ser divididas nas seguintes:

- Funções de Conversão de Maiúsculas e Minúsculas
- Funções de manipulação de caractere

Função	Objetivo
LOWER(<i>coluna/expressão</i>)	Converte valores de caractere alfabético para letras minúsculas
UPPER(<i>coluna/expressão</i>)	Converte valores de caractere alfabético para letras maiúsculas
INITCAP(<i>coluna/expressão</i>)	Converte valores de caractere alfabético para usar maiúscula na primeira letra de cada palavra e todas as outras letras em minúsculas
CONCAT(<i>coluna1/expressão1, coluna2/expressão2</i>)	Concatena o primeiro valor do caractere ao segundo valor do caractere, equivalente ao operador de concatenação ()
SUBSTR(<i>coluna/expressão,m[,n]</i>)	Retorna caracteres específicos a partir do valor de caractere começando na posição <i>m</i> , até <i>n</i> caracteres depois (Se <i>m</i> for negativo, a conta inicia no final do valor de caractere. Se <i>n</i> for omitido, são retornados todos os caracteres até o final da string.)

Observação: Essas funções abordadas nesta lição são um subconjunto de funções disponíveis.

Funções de Caractere



Funções de Caractere (continuação)

Função	Objetivo
LENGTH(<i>coluna/expressão</i>)	Retorna o número de caracteres do valor
INSTR(<i>coluna/expressão,m</i>)	Retorna a posição numérica do caractere nomeado
LPAD(<i>coluna/expressão, n, 'string'</i>)	Preenche o valor de caracter justificado à direita a uma largura total de <i>n</i> posições de caractere
TRIM(<i>anterior/posterior/ambos, trim_character FROM trim_source</i>)	Permite a você organizar cabeçalhos ou caracteres de fim de linha (ou os dois) a partir de uma string de caractere. Se <i>trim_character</i> ou <i>trim_source</i> for um caractere literal, você deve incluí-los entre aspas simples. Este é um recurso disponível a partir Oracle8i em diante.

Funções de Conversão de Maiúsculas e Minúsculas

- Converter maiúsculas em minúsculas para strings de caractere

Função	Resultado
LOWER('SQL Course')	sql course
UPPER('SQL Course')	SQL COURSE
INITCAP('SQL Course')	Sql Course

Funções de Conversão de Maiúsculas e Minúsculas

LOWER, UPPER e INITCAP são as três funções de conversão de maiúscula em minúscula.

LOWER: Converte string de caracteres com letras maiúsculas e minúsculas ou só maiúsculas para letras minúsculas

UPPER: Converte string de caracteres com letras maiúsculas e minúsculas ou só minúsculas para letras maiúsculas

INITCAP: Converte a primeira letra de cada palavra para maiúscula e mantém as outras letras em minúscula

```
SQL> SELECT 'The job title for '||INITCAP(ename)||' is '  
2  ||LOWER(job) AS "EMPLOYEE DETAILS"  
3  FROM emp;
```

EMPLOYEE DETAILS

The job title for King is president

The job title for Blake is manager

The job title for Clark is manager

...

14 rows selected.

Usando Funções de Conversão de Maiúsculas e Minúsculas

- Exibir o número de funcionário, nome e número de departamento do funcionário Blake.

```
SQL> SELECT empno, ename, deptno
2 FROM emp
3 WHERE ename = 'blake';
no rows selected
```

```
SQL> SELECT empno, ename, deptno
2 FROM emp
3 WHERE ename = UPPER('blake');
```

EMPNO	ENAME	DEPTNO
7698	BLAKE	30

Funções de Conversão de Maiúsculas e Minúsculas (continuação)

O exemplo do slide exibe o número de funcionário, nome e número de departamento do funcionário BLAKE.

A cláusula WHERE da primeira instrução SQL especifica o nome do funcionário como 'blake'. Visto que todos os dados da tabela EMP são armazenados em letra maiúscula, o nome 'blake' não localiza uma correspondência na tabela EMP e como resultado nenhuma linha será selecionada.

A cláusula WHERE da segunda instrução SQL especifica que o nome do funcionário na tabela EMP seja comparado a 'blake', convertido para letras maiúsculas. Agora que os dois nomes estão em maiúscula, é localizada uma correspondência e uma linha é selecionada. A cláusula WHERE pode ser escrita novamente da seguinte forma para produzir o mesmo resultado:

```
... WHERE ename = 'BLAKE'
```

O nome na entrada aparece como se estivesse armazenado no banco de dados. Para exibir o nome com a primeira letra maiúscula, use a função INITCAP na instrução SELECT.

```
SQL> SELECT empno, INITCAP(ename), deptno
2 FROM emp
3 WHERE ename = UPPER('blake');
```

Funções de Manipulação de Caractere

- Manipular strings de caractere

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

Funções de Manipulação de Caractere

CONCAT, SUBSTR, LENGTH, INSTR, LPAD e TRIM são as seis funções de manipulação de caractere abordadas nesta lição.

CONCAT: Une valores de junção (Você está limitado a usar dois parâmetros com CONCAT.)

SUBSTR: Extrai uma string de determinado tamanho

LENGTH: Exibe o tamanho de uma string como um valor numérico

INSTR: Localiza a posição numérica do caractere nomeado

LPAD: Preenche o valor do caractere justificado à direita

Observação: A função de manipulação de caractere **RPAD** preenche o valor de caractere justificado à esquerda

TRIM: Organiza cabeçalho ou caracteres de fim de linha (ou os dois) a partir de uma string de caractere. Se trim_character ou trim_source forem um caractere literal, você deve incluí-los entre aspas simples.

Usando as Funções de Manipulação de Caractere

```
SELECT ename, CONCAT (ename, job), LENGTH(ename),  
        INSTR(ename, 'A')  
FROM    emp  
WHERE   SUBSTR(job,1,5) = 'SALES';
```

ENAME	CONCAT (ENAME, JOB)	LENGTH (ENAME)	INSTR (ENAME, 'A')
MARTIN	MARTINSALESMAN	6	2
ALLEN	ALLENSALESMAN	5	1
TURNER	TURNERSALESMAN	6	0
WARD	WARDSALESMAN	4	2



Funções de Manipulação de Caractere (continuação)

O exemplo do slide exibe o nome do funcionário e o cargo juntos, o tamanho do nome e a posição numérica da letra A no nome do funcionário, para todos os funcionários que estão em vendas.

Exemplo

Modifique a instrução SQL no slide a fim de exibir os dados daqueles funcionários cujos nomes terminam com N.

```
SQL> SELECT  ename, CONCAT(ename, job), LENGTH(ename),  
2           INSTR(ename, 'A')  
3 FROM      emp  
4 WHERE     SUBSTR(ename, -1, 1) = 'N';
```

ENAME	CONCAT (ENAME, JOB)	LENGTH (ENAME)	INSTR (ENAME, 'A')
MARTIN	MARTINSALESMAN	6	2
ALLEN	ALLENSALESMAN	5	1

Funções Numéricas

- ROUND: Arredonda valor para determinado decimal

ROUND(45.926, 2)  45.93

- TRUNC: Trunca valor para determinado decimal

TRUNC(45.926, 2)  45.92

- MOD: Retorna o restante da divisão

MOD(1600, 300)  100

Funções Numéricas

As **funções numéricas** aceitam entrada numérica e retornam valores numéricos. Esta seção descreve algumas das funções numéricas.

Função	Objetivo
ROUND(<i>coluna</i> <i>expressão</i> , <i>n</i>)	Arredonda a coluna, expressão ou valor para <i>n</i> casas decimais ou se <i>n</i> for omitido, nenhuma casa decimal (Se <i>n</i> for negativo, os números à esquerda do ponto decimal serão arredondados.)
TRUNC(<i>coluna</i> <i>expressão</i> , <i>n</i>)	Trunca a coluna, expressão ou valor para <i>n</i> casas decimais ou se <i>n</i> for omitido, nenhuma casa decimal (Se <i>n</i> for negativo, os números à esquerda do ponto decimal serão truncados para zero.)
MOD(<i>m</i> , <i>n</i>)	Retorna o resto de <i>m</i> dividido por <i>n</i> .

Observação: Esta lista é um subconjunto das funções numéricas disponíveis. Para obter mais informações, consulte o *Oracle Server SQL Reference*, "Number Functions".

Usando a Função ROUND

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

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

Função ROUND

A função **ROUND** arredonda a coluna, expressão ou valor para *n* casas decimais. Se o segundo argumento for 0 ou estiver ausente, o valor será arredondado para nenhuma casa decimal. Se o segundo argumento for 2, o valor será arredondado para duas casas decimais. Da mesma forma, se o segundo argumento for -2, o valor será arredondado para duas casas decimais para a esquerda.

A função ROUND pode também ser utilizada com funções de data. Você verá exemplos mais adiante nesta seção.

A tabela DUAL é fictícia. Mais informações serão fornecidas mais tarde.

Usando a Função TRUNC

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

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

Função TRUNC

A função **TRUNC** trunca a coluna, expressão ou valor para n casas decimais.

A função TRUNC trabalha com argumentos similares aos da função ROUND. Se o segundo argumento for 0 ou estiver ausente, o valor será truncado para nenhuma casa decimal. Se o segundo argumento for 2, o valor será truncado para duas casas decimais. Da mesma forma, se o segundo argumento for -2, o valor será truncado para duas casas decimais para esquerda.

Semelhante à função ROUND, a função TRUNC pode ser usada com funções de data.

Usando a Função MOD

- Calcular o restante da proporção do salário para comissão de todos os funcionários cujo cargo é salesman.

```
SELECT  ename, sal, comm, MOD(sal, comm)
FROM    emp
WHERE   job = 'SALESMAN';
```

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

Função MOD

A função **MOD** localiza o restante do valor 1 dividido pelo valor 2. O exemplo do slide calcula o restante da proporção de salário para comissão de todos os funcionários cujo cargo é salesman.

Trabalhando com Datas

- O Oracle armazena datas em um formato numérico interno: século, ano, mês, dia, horas, minutos, segundos.
- O formato de data default é DD-MON-YY.
- SYSDATE é uma função de retorno de data e hora.
- DUAL é uma tabela fictícia usada para visualizar SYSDATE.

Formato de Data Oracle

Os produtos da Oracle armazenam **datas** em um formato numérico interno, representando o século, ano, mês, dia, horas, minutos e segundos.

O formato de entrada e exibição default para qualquer data é **DD-MON-YY**. Datas válidas para a Oracle estão entre 1 de janeiro, 4712 A.C. e 31 de dezembro, 9999 D.C.

SYSDATE

SYSDATE é a função de data que retorna a data e a hora atual. Você pode usar o SYSDATE da mesma forma como usaria qualquer outro nome de coluna. Por exemplo, é possível exibir a data atual selecionando SYSDATE a partir da tabela. Costuma-se selecionar SYSDATE em uma tabela fictícia chamada DUAL.

DUAL

A tabela DUAL pertence ao usuário SYS e pode ser acessada por todos os usuários. Ela contém uma coluna, DUMMY, e uma linha com o valor X. A tabela DUAL é útil quando deseja retornar um valor somente uma vez — por exemplo, o valor de uma constante, pseudocoluna ou expressão que não é derivada de uma tabela com dados do usuário. A tabela DUAL, em geral, é utilizada pela totalidade de sintaxe da cláusula SELECT, pois as duas cláusulas SELECT e FROM são obrigatórias e diversos cálculos não precisam selecionar das tabelas reais.

Exemplo

Exibir a data atual usando a tabela DUAL.

```
SQL> SELECT    SYSDATE  
      2 FROM    DUAL;
```

Aritmética com Datas

- Adicionar ou subtrair um número de, ou para, uma data para um valor de *data* resultante.
- Subtrair duas datas a fim de localizar o *número* de dias entre estas datas.
- Adicionar *horas* para uma data dividindo o número de horas por 24.



Aritmética com Datas

Já que os bancos de dados armazenam datas como números, você pode executar cálculos usando operadores aritméticos como adição e subtração. É possível adicionar e subtrair constantes de número bem como datas. Você pode executar 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 horas para uma data

Usando Operadores Aritméticos com Datas

```
SELECT ename, (SYSDATE-hiredate)/7 WEEKS  
FROM    emp  
WHERE   deptno = 10;
```

ENAME	WEEKS
-----	-----
KING	830.93709
CLARK	853.93709
MILLER	821.36566

Aritmética com Datas (continuação)

O exemplo no slide exibe o nome e o número de semanas empregadas por todos os funcionários do departamento 10. Ele subtrai a data atual (SYSDATE) a partir da data na qual o funcionário foi admitido e divide o resultado por 7 a fim de calcular o número de semanas que o trabalhador está empregado.

Observação: SYSDATE é uma função SQL que retorna a data e a hora atual. Seus resultados podem ser diferentes dos obtidos no exemplo.

Funções de Data

Função	Descrição
MONTHS_BETWEEN	Número de meses entre duas datas
ADD_MONTHS	Adiciona meses de calendário para a data
NEXT_DAY	Dia seguinte da data especificada
LAST_DAY	Último dia do mês
ROUND	Data de arredondamento
TRUNC	Data para truncada

Funções de Data

Funções de data operam em datas Oracle. Todas as funções de data retornam um valor de tipo de dados DATE exceto MONTHS_BETWEEN, que retorna um valor numérico.

MONTHS_BETWEEN(*data1*, *data2*): Localiza o número de meses entre a *data 1* e a *data 2*. O resultado pode ser positivo ou negativo. Se *data1* for posterior a *data2*, o resultado será positivo; se *data1* for anterior a *data2*, o resultado será negativo. A parte não-inteira do resultado representa uma parte do mês.

ADD_MONTHS(*data*, *n*): Adiciona um número *n* de meses de calendário à *data*. O valor de *n* deve ser inteiro ou pode ser negativo.





NEXT_DAY(*data*, '*char*'): Localiza a data do próximo dia especificado da *data* seguinte da semana ('*char*'). O valor de *char* pode ser um número representando um dia ou uma string de caractere.

LAST_DAY(*data*): Localiza a data do último dia do mês que contém a *data*.

ROUND(*data*[, '*fmt*']): Retorna a *data* arredondada para a unidade especificada pelo modelo de formato *fmt*. Se o modelo de formato *fmt* for omitido, a *data* será arredondada para o dia mais próximo.

TRUNC(*data*[, '*fmt*']): Retorna a *data* com a parte da hora do dia truncada para a unidade especificada pelo modelo de formato *fmt*. Se o modelo de formato *fmt* for omitido, a *data* será truncada para o dia mais próximo.

Usando 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'**

Funções de Data (continuação)

Para todos os funcionários contratados por menos de 200 meses, exibir o número do funcionário, a data de admissão, o número de meses contratado, a data de revisão semestral, a primeira sexta-feira após a data da admissão e o último dia do mês em que foi admitido.

```
SQL> SELECT empno, hiredate,
2      MONTHS_BETWEEN(SYSDATE, hiredate) TENURE,
3      ADD_MONTHS(hiredate, 6) REVIEW,
4      NEXT_DAY(hiredate, 'FRIDAY'),
LAST_DAY(hiredate)
5 FROM emp
6 WHERE MONTHS_BETWEEN (SYSDATE,
hiredate)<200;
```

```
EMPNO HIREDATE      TENURE REVIEW      NEXT_DAY( LAST_DAY(  -----
- - - - -
7839 17-NOV-81  192.24794 17-MAY-82  20-NOV-81  30-NOV-81
7698 01-MAY-81  198.76407 01-NOV-81  08-MAY-81  31-MAY-81
...
11 rows selected.
```

Usando 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 Data (continuação)

As funções ROUND e TRUNC podem ser usadas para os valores de data e número. Quando usadas com datas, estas funções arredondam ou truncam o modelo de formato especificado. Assim, é possível arredondar datas para o ano ou mês mais próximo.

Exemplo

Compare as datas de admissão de todos os funcionários contratados em 1982. Exiba o número do funcionário, a data de admissão e o mês de início usando as funções ROUND e TRUNC.

```
SQL> SELECT empno, hiredate,
2          ROUND(hiredate, 'MONTH'),
TRUNC(hiredate, 'MONTH')
3 FROM emp
4 WHERE hiredate like '%1982';
```

EMPNO	HIREDATE	ROUND(HIR	TRUNC(HIR
7788	09-DEC-82	01-DEC-82	01-DEC-82
7934	23-JAN-82	01-FEB-82	01-JAN-82

Funções de Conversão



Funções de Conversão

Além dos tipos de dados Oracle, as colunas de tabela em um banco de dados podem ser definidas usando os tipos de dados ANSI, DB2 e SQL/DS. No entanto, o Oracle Server converte internamente tais tipos de dados para tipos de dados Oracle.

Em alguns casos, o Oracle Server aceita dados de um tipo de dados em que espera dados de um tipo de dados diferente. Isso é permitido quando o Oracle Server pode converter dados automaticamente para o tipo de dados esperado. Essa **conversão de tipo de dados** pode ser realizada de forma *implícita* pelo Oracle Server ou *explícita* pelo usuário.

As conversões implícitas de tipo de dados funcionam de acordo com as regras explicadas nos próximos dois slides.

As conversões explícitas de tipo de dados podem ser realizadas usando-se as funções de conversão. As funções de conversão convertem um valor de um tipo de dados para outro. Geralmente, o formato dos nomes de função seguem a convenção *tipo de dados TO tipo de dados*. O primeiro tipo de dados é um tipo de dados de entrada; o último tipo de dados é de saída.

Observação: Embora a conversão implícita de tipos de dados esteja disponível, é recomendável que você realize a conversão explícita de tipo de dados a fim de garantir a confiabilidade das instruções SQL.

Conversão Implícita de Tipo de Dados

- Para atribuições, o Oracle Server pode converter automaticamente o seguinte:

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

Conversão Implícita de Tipo de Dados

A atribuição ocorre se o Oracle Server puder converter o tipo de dados do valor nela usado para a atribuição destino.

Conversão Implícita de Tipo de Dados

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

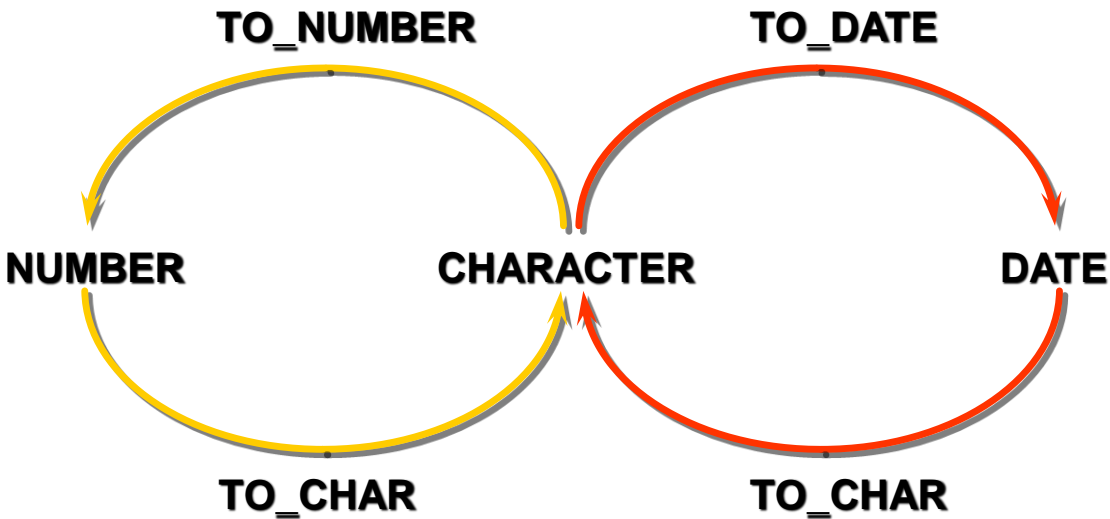
De	Para
VARCHAR2 ou CHAR	NUMBER
VARCHAR2 ou CHAR	DATE

Conversão Implícita de Tipo de Dados

Em geral, o Oracle Server usa a regra para expressão quando a conversão de tipo de dados é necessária em lugares não cobertos por uma regra para as conversões de atribuição.

Observação: 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 DD-MON-YY.

Conversão Explícita de Tipo de Dados

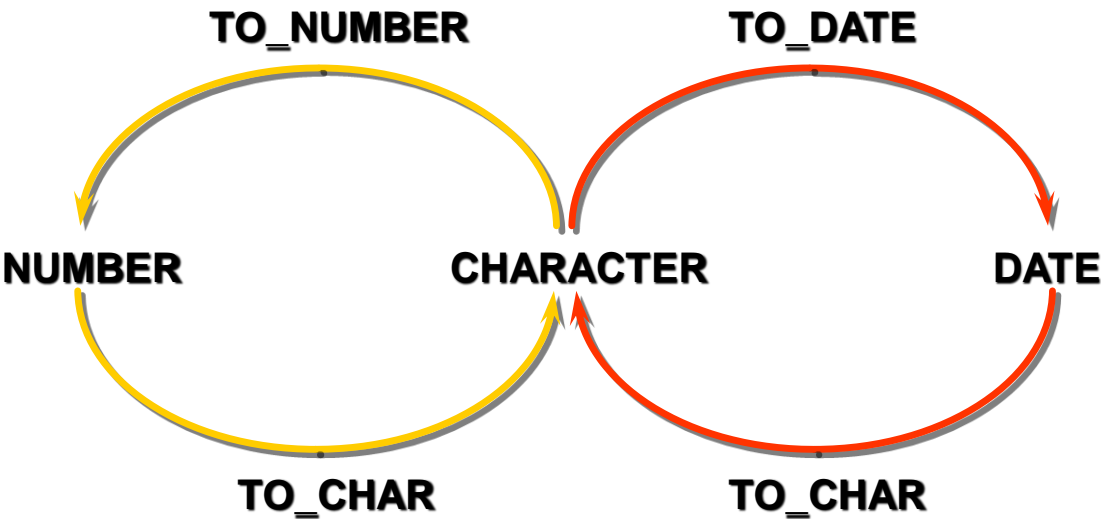


Conversão Explícita de Tipo de Dados

O SQL oferece três funções para converter um valor de um tipo de dados para outro:

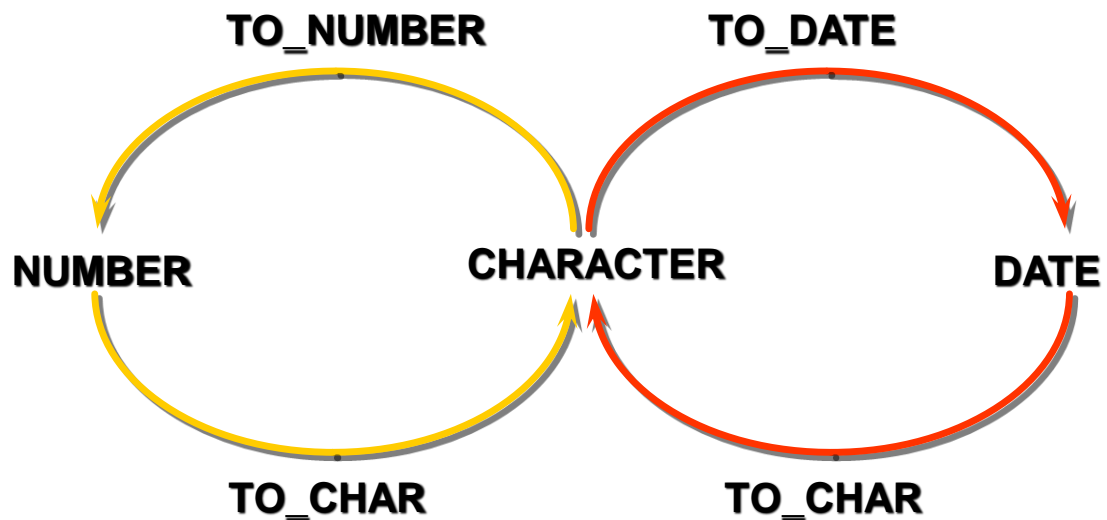
Função	Objetivo
<code>TO_CHAR(número data,[fmt],[nlsparams])</code>	<p>Converte um valor de número ou data para uma string de caractere VARCHAR2 com modelo de formato <i>fmt</i>.</p> <p>Conversão de Número:</p> <p>O parâmetro <i>nlsparams</i> especifica os seguintes caracteres, que são retornados por elementos de formato de número:</p> <ul style="list-style-type: none">• Caractere decimal• Separador de grupo• Símbolo da moeda local• Símbolo da moeda internacional <p>Se <i>nlsparams</i> ou qualquer outro parâmetro for omitido, essa função utilizará os valores de parâmetro default para a seção.</p>

Conversão Explícita de Tipo de Dados



Função	Objetivo
TO_CHAR(<i>número</i> <i>data</i> , [<i>fmt</i>], [<i>nlsparms</i>])	Conversão de data: O parâmetro <i>nlsparms</i> especifica o idioma no qual os nomes de dias, meses e abreviação retornam. Se este parâmetro for omitido, essa função utilizará os idiomas de data default para a seção.
TO_NUMBER(<i>carac</i> , [<i>fmt</i>], [<i>nlsparms</i>])	Converte uma string de caractere contendo dígitos para um número no formato especificado pelo modelo de formato opcional <i>fmt</i> . O parâmetro <i>nlsparms</i> tem a mesma finalidade nesta função como na função TO_CHAR para a conversão de número.
TO_DATE(<i>carac</i> , [<i>fmt</i>], [<i>nlsparms</i>])	Converte uma string de caractere representando uma data para um valor de data de acordo com o <i>fmt</i> especificado. Se <i>fmt</i> for omitido, o formato é DD-MON-YY. O parâmetro <i>nlsparms</i> possui a mesma finalidade na função TO_CHAR para a conversão de data.

Conversão Explícita de Tipo de Dados



Conversão Explícita de Tipo de Dados (continuação)

Observação: A lista de funções mencionadas nesta lição é um subconjunto das funções de conversão disponíveis.

Para obter mais informações, consulte o *Oracle8 Server SQL Reference*, "Conversion Functions".

Função TO_CHAR com Datas

```
TO_CHAR(data, 'fmt')
```

- O modelo de formato:
 - Deve estar entre aspas simples 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

Exibindo uma Data em um Formato Específico

Anteriormente, todos os valores de data Oracle eram exibidos no formato DD-MON-YY. A função TO_CHAR permite converter uma data a partir desse formato default para um especificado por você.

Diretrizes

O **modelo de formato** deve estar entre aspas simples e fazer distinção entre maiúsculas e minúsculas.

O modelo de formato pode incluir qualquer elemento de formato de data válido. Certifique-se de separar o valor da data do modelo de formato por uma vírgula.

Os nomes de dias e meses na saída são preenchidos automaticamente por espaços.

Para remover os espaços em branco ou suprimir os zeros à esquerda, use o modo de preenchimento do elemento **fm**.

Você pode redimensionar o tamanho da exibição do campo de caractere resultante com o comando COLUMN do SQL*Plus.

A largura da coluna resultante é, por padrão, de 80 caracteres.

```
SQL> SELECT empno, TO_CHAR(hiredate, 'MM/YY') Month_Hired
2 FROM emp
3 WHERE ename = 'BLAKE';
```

Elementos de Modelo de Formato de Data

YYYY	Ano completo em números
YEAR	Ano por extenso
MM	Valor de dois dígitos para mês
MONTH	Abreviação de três letras do dia da semana
DY	Abreviação de três letras do dia da semana
DAY	Nome completo do dia

Elementos de Exemplo de Formatos de Data Válidos

Elemento	Descrição
SCC ou CC	Século; Prefixos S data AC com -
Anos em datas YYYY ou SYYYY	Ano; Prefixos S data AC com -
YYY ou YY ou Y	Últimos três, dois ou um dígitos do ano
Y,YYY	Ano com vírgula nesta posição
IYYY, IYY, IY, I	Quatro, três, dois ou um dígito do ano com base no padrão ISO
SYEAR ou YEAR	Ano inteiro; Prefixos S data AC com -
BC ou AD	Indicador AC/DC
B.C. ou A.D.	Indicador com pontos AC/DC
Q	Trimestre do ano
MM	Mês, valor de dois dígitos
MONTH	Nome do mês preenchido com espaços limitado a nove caracteres
MON	Nome do mês, abreviação de três letras
RM	Mês em números romanos
WW ou W	Semana do ano ou mês
DDD ou DD ou D	Dia do ano, mês ou semana
DAY	Nome do dia preenchido com espaços limitado a nove caracteres
DY	Nome do dia; abreviação de três letras
J	Dia do calendário juliano; o número de dias desde 31 de dezembro 4713 BC

Elementos de Modelo de Formato de Data

- Elementos de hora formatam a parte de hora da data.

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

- Adicionar strings de caractere incluindo-as entres aspas.

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

- Sufixos de número escrevem os números por extenso.

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

Formatos de Hora

Use os formatos listados tabelas a seguir para exibir informações de hora e literais e alterar numerais para números escritos.

Elemento	Descrição
AM ou PM	Indicador meridiano
A.M. ou P.M.	Indicador meridiano com pontos
HH ou HH12 ou HH24	Horas do dia ou hora (1 a 12) ou hora (0 a 23)
MI	Minuto (0 a 59)
SS	Segundo (0 a 59)
SSSS	Segundos após a meia-noite (0–86399)

Outros Formatos

Elemento	Descrição
/ . ,	A pontuação é reproduzida no resultado
"of the"	A string entre aspas é reproduzida no resultado

Especificando Sufixos para Influenciar Exibição de Número

Elemento	Descrição
TH	Número ordinal (por exemplo, DDTH para 4TH)
SP	Número por extenso (por exemplo, DDSP para FOUR)
SPTH ou THSP	Números ordinais por extenso (por exemplo, DDSPTH para FOURTH)

Usando a Função TO_CHAR com Datas

```
SQL> SELECT ename,  
2         TO_CHAR(hiredate, 'fmDD Month YYYY') HIREDATE  
3 FROM emp;
```

ENAME	HIREDATE
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 Datas

A instrução SQL no slide exibe o nome e as datas de admissão para todos os funcionários. A data de admissão aparece como 17 November 1981.

Exemplo

Modificar o exemplo do slide para exibir as datas em um formato que apareça como Seventh of February 1981 08:00:00 AM.

```
SQL> SELECT ename,  
2         TO_CHAR(hiredate, 'fmDdspth "of" Month YYYY fmHH:MI:SS AM')  
3         HIREDATE  
4 FROM emp;
```

ENAME	HIREDATE
KING	Seventeenth of November 1981 12:00:00 AM
BLAKE	First of May 1981 12:00:00 AM
...	

14 rows selected.

Note que o mês segue o formato do modelo de formato especificado, ou seja, a primeira letra em maiúscula e o restante em minúscula.

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:

9	Representa um número
0	Forces a zero to be displayed
\$	Coloca um sinal de dólar flutuante
L	Usa o símbolo da moeda local flutuante
.	Imprime um ponto decimal
,	Imprime um indicador de milhar

Função TO_CHAR com Números

Ao trabalhar com valores de número tais como strings de caractere, você deve converter esse números para o tipo de dados de caractere usando a função **TO_CHAR**, que traduz o tipo de dados de NUMBER para tipo de dados VARCHAR2. Essa técnica é particularmente útil com concatenação.

Elementos de Formato de Número

Se estiver convertendo um número para tipo de dados de caractere, você pode usar os seguintes elementos:

Elemento	Descrição	Exemplo	Resultado
9	Posição numérica (número de 9s determinam o tamanho da exibição)	999999	1234
0	Exibe zeros à esquerda	099999	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
MI	Sinais de menos à direita (valores negativos)	999999MI	1234-
PR	Coloca números negativos entre parênteses	999999PR	<1234>
EEEE	Notação científica (formato deve especificar quatro Es)	99.999EEEE	1.234E+03
V	Multiplica por 10 <i>n</i> vezes (<i>n</i> = número de 9s após o V)	9999V99	123400
B	Exibe valores de zero como espaço, não 0	B9999.99	1234.00

Usando a Função TO_CHAR com Números

```
SELECT      TO_CHAR(sal, '$99,999') SALARY
FROM        emp
WHERE       ename = 'SCOTT';
```

SALARY

\$3,000

Diretrizes

O Oracle Server exibe uma string com sinais numéricos (#) no lugar de um número inteiro cujos dígitos excedam o número de dígitos fornecidos no modelo de formato.

O Oracle Server arredonda os valores decimais armazenados para o número de espaços decimais fornecidos no modelo de formato.

Funções TO_NUMBER e TO_DATE

- Converter uma string de caractere para um formato de número usando a função TO_NUMBER

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

- Converter uma string de caractere para um formato de data usando a função TO_DATE

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

Funções TO_NUMBER e TO_DATE

Talvez deseje converter uma string de caractere para um número ou data. Para completar essa tarefa, use as funções **TO_NUMBER** ou **TO_DATE**. O modelo de formato que escolher será baseado nos elementos de formato demonstrados anteriormente.

Exemplo

Exibir os nomes e as datas de admissão para todos os funcionários contratados em 22 de fevereiro de 1981.

```
SQL> SELECT ename, hiredate
2 FROM emp
3 WHERE hiredate = TO_DATE('February 22, 1981', 'Month dd, YYYY');
```

ENAME	HIREDATE
WARD	22-FEB-81

Função NVL

- Converte nulo para um valor real
 - Os tipos de dados que podem ser usados são data, caractere e número.
 - Os tipos de dados devem corresponder com
 - NVL(comm,0)
 - NVL(hiredate,'01-JAN-97')
 - NVL(job,'No Job Yet')

A Função NVL

Para converter um valor nulo para um valor real, use a função NVL.

Syntax

NVL (expr1, expr2)

onde: *expr1* é o valor de origem ou expressão que pode conter nulo
expr2 é o valor de destino para a conversão de nulo

Você pode usar a função NVL para converter qualquer tipo de dados, porém o valor de devolução é sempre o mesmo do tipo de dados da *expr1*.

Conversões NVL para Vários Tipos de Dados

Tipo de dados	Exemplo de Conversão
NUMBER	NVL(<i>number_column</i> ,9)
DATE	NVL(<i>date_column</i> , '01-JAN-95')
CHAR ou VARCHAR2	NVL(<i>character_column</i> , 'Unavailable')

Usando a Função NVL

```
SELECT ename, sal, comm, (sal*12)+NVL(comm,0)
FROM    emp;
```

ENAME	SAL	COMM	(SAL*12)+NVL(COMM,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 NVL

Para calcular a remuneração anual de todos os funcionários, você precisa multiplicar o salário mensal por 12 e, em seguida, adicionar a comissão.

```
SQL> SELECT ename, sal, comm, (sal*12)+comm
2 FROM emp;
```

ENAME	SAL	COMM	(SAL*12)+COMM
-----	-----	-----	-----
KING	5000		
BLAKE	2850		
CLARK	2450		
JONES	2975		
MARTIN	1250	1400	16400
...			
14 rows selected.			

Note que a remuneração anual é calculada somente para aqueles funcionários que recebem uma comissão. Se qualquer valor de coluna na expressão for nulo, o resultado será nulo. Para calcular valores para todos os funcionários, você deve converter o valor nulo para um número antes de aplicar o operador aritmético. No exemplo do slide, a função para NVL é usada para converter valores nulos para zero.

Função DECODE

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

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

A Função DECODE

A função **DECODE** decodifica uma expressão de um modo similar à lógica IF-THEN-ELSE usada em diversas linguagens. A função DECODE decodifica a *expressão* após compará-la a cada valor de *pesquisa*. Se a expressão for a mesma da *pesquisa*, o *resultado* é retornado.

Se o valor default for omitido, será retornado um valor nulo onde um valor de pesquisa não corresponde a quaisquer valores de resultado.

Usando a Função DECODE

```
SELECT job, sal,
       DECODE(job, 'ANALYST', SAL*1.1,
               'CLERK',   SAL*1.15,
               'MANAGER', SAL*1.20,
               SAL)
       REVISSED_SALARY
FROM   emp;
```

JOB	SAL	REVISSED_SALARY
PRESIDENT	5000	5000
MANAGER	2850	3420
MANAGER	2450	2940
...		

14 rows selected.

Usando a Função DECODE

Na instrução SQL acima, o valor de JOB está decodificado. Se JOB for ANALYST, o aumento de salário é de 10%; se JOB for CLERK, o aumento de salário é de 15%; se JOB for MANAGER, o aumento de salário é de 20%. Para todas as outras funções de trabalho, não há aumento de salário.

A mesma instrução pode ser escrita como uma instrução IF-THEN-ELSE:

```
IF job = 'ANALYST' THEN sal = sal*1.1
IF job = 'CLERK'   THEN sal = sal*1.15
IF job = 'MANAGER' THEN sal = sal*1.20
ELSE sal = sal
```


Usando a Função DECODE

- Exibir a taxa de imposto aplicável para cada funcionário do departamento 30.

```
SELECT ename, sal,
        DECODE (TRUNC (sal/1000, 0),
                0, 0.00,
                1, 0.09,
                2, 0.20,
                3, 0.30,
                4, 0.40,
                5, 0.42,
                6, 0.44,
                0.45) TAX_RATE
FROM    emp
WHERE   deptno = 30;
```

Exemplo

O slide mostra outro exemplo usando a função DECODE. Neste exemplo, determinamos a taxa de imposto para cada funcionário do departamento 30 com base no salário mensal. As taxas de imposto são mencionadas por valores na tabela a seguir.

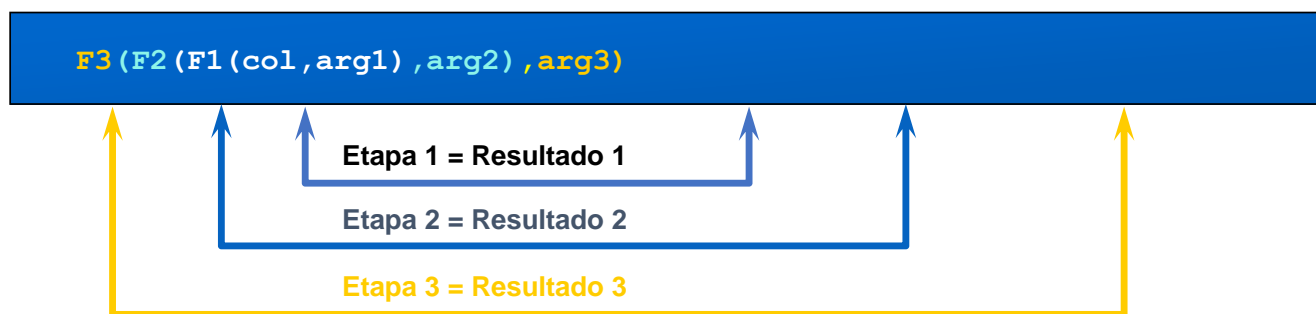
<i>Faixa Salarial Mensal</i>	<i>Taxa</i>
US\$0,00 - 999,99	0%
US\$1.000,00 - 1.999,99	9%
US\$2.000,00 - 2.999,99	20%
US\$3.000,00 - 3.999,99	30%
US\$4.000,00 - 4.999,99	40%
US\$5.000,00 - 2.999,99	42%
US\$6.000,00 - 6.999,99	44%
US\$7.000,00 ou superior	45%

ENAME	SAL	TAX_RATE
-----	-----	-----
BLAKE	2850	.2
MARTIN	1250	.09
ALLEN	1600	.09
TURNER	1500	.09

...
6 rows selected.

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.



Aninhando Funções

As funções de uma única linha podem ser aninhadas em qualquer nível. As **funções aninhadas** são avaliadas desde o nível mais interno até o mais externo. Alguns exemplos a seguir mostram a flexibilidade dessas funções.

Aninhando Funções

```
SELECT      ename,  
            NVL(TO_CHAR(mgr), 'No Manager')  
FROM        emp  
WHERE       mgr IS NULL;
```

ENAME	NVL(TO_CHAR(MGR), 'NOMANAGER')
-----	-----
KING	No Manager

Aninhando Funções (continuação)

O exemplo do slide exibe o presidente da empresa, que não possui nenhum gerente.

A avaliação da instrução SQL envolve duas etapas:

1. Avaliar a função interna a fim de converter um valor de número para uma string de caractere.

Result1 = TO_CHAR(mgr)

2. Avaliar a função externa a fim de substituir o valor nulo com uma string de texto.

NVL(Result1, 'No Manager')

A expressão inteira transforma-se no cabeçalho da coluna, pois não foi fornecido nenhum apelido de coluna.

Exemplo

Exibir a data da próxima sexta-feira, que fica seis meses após a data de admissão. A data resultante deve aparecer como Friday, March 12th, 1982. Ordenar os resultados por data de admissão.

```
SQL> SELECT      TO_CHAR(NEXT_DAY(ADD_MONTHS  
2              (hiredate, 6), 'FRIDAY'),  
3              'fmDay, Month ddth, YYYY')  
4              "Next 6 Month Review"  
5 FROM          emp  
6 ORDER BY      hiredate;
```

Sumário

- Use as funções para realizar o seguinte:
 - Executar cálculos usando dados
 - Modificar itens de dados individuais
 - Manipular saída para grupos de linhas
 - Alterar formatos de data para exibição
 - Converter tipos de dados de coluna

Funções de Uma Única Linha

As funções de uma única linha podem ser aninhadas em qualquer nível. As funções de uma única linha podem manipular o seguinte:

Dados de caractere: LOWER, UPPER, INITCAP, CONCAT, SUBSTR, INSTR, LENGTH

Dados de número: ROUND, TRUNC, MOD

Dados de data: MONTHS_BETWEEN, ADD_MONTHS, NEXT_DAY, LAST_DAY, ROUND, TRUNC

Valores de data podem também usar operadores aritméticos.

As funções de conversão podem converter valores de caractere, de data e numérico: TO_CHAR, TO_DATE, TO_NUMBER

SYSDATE e DUAL

SYSDATE é uma função de data que retorna a data e a hora atual. Costuma-se selecionar SYSDATE em uma tabela fictícia chamada DUAL.

Visão Geral do Exercício

- Criando consultas que exigem o uso de funções numéricas, de caractere e de data
- Usando concatenação com funções
- Criando consultas que diferenciam entre letras maiúsculas e minúsculas para testar a utilidade das funções de caractere
- Executando cálculos de anos e meses de serviço de um funcionário
- Determinando a data de revisão para um funcionário



Visão Geral do Exercício

Este exercício é desenvolvido para oferecer a você uma variedade de exercícios usando diferentes funções disponíveis para tipos de dados de caractere, de número e data.

Lembre-se que para funções aninhadas, os resultados são avaliados a partir da função mais interna para a função mais externa.

Exercício 3

1. Crie uma consulta para exibir a data atual. Coloque um label na coluna Date.

```
Date
-----
28-OCT-97
```

2. Exiba o número do funcionário, o nome, o salário e o aumento salarial de 15% expresso como número inteiro. Coloque um label na coluna New Salary. Salve a instrução SQL em um arquivo nomeado p3q2.sql.
3. Execute a consulta no arquivo p3q2.sql.

EMPNO	ENAME	SAL	New Salary
-----	-----	-----	-----
7839	KING	5000	5750
7698	BLAKE	2850	3278
7782	CLARK	2450	2818
7566	JONES	2975	3421
7654	MARTIN	1250	1438
7499	ALLEN	1600	1840
7844	TURNER	1500	1725
7900	JAMES	950	1093
7521	WARD	1250	1438
7902	FORD	3000	3450
7369	SMITH	800	920
7788	SCOTT	3000	3450
7876	ADAMS	1100	1265
7934	MILLER	1300	1495

14 rows selected.

4. Modifique a consulta p3q2.sql para adicionar uma coluna que subtrairá o salário antigo do novo salário. Coloque um label na coluna Increase. Execute novamente a consulta.

EMPNO	ENAME	SAL	New Salary	Increase
-----	-----	-----	-----	-----
7839	KING	5000	5750	750
7698	BLAKE	2850	3278	428
7782	CLARK	2450	2818	368
7566	JONES	2975	3421	446
7654	MARTIN	1250	1438	188
7499	ALLEN	1600	1840	240
7844	TURNER	1500	1725	225
7900	JAMES	950	1093	143
.

14 rows selected.

Exercício 3 (continuação)

5. Exibe o nome do funcionário, a data de admissão, que é a primeira segunda-feira após seis meses de serviço. Coloque um label na coluna REVIEW. Formate as datas que aparecem em formato semelhante a "Sunday, the Seventh of September, 1981".

	ENAME	HIREDATE	REVIEW
	KING	17-NOV-81	Monday, the Twenty-Fourth of May, 1982
	BLAKE	01-MAY-81	Monday, the Second of November, 1981
1981	CLARK	09-JUN-81	Monday, the Fourteenth of December,
	JONES	02-APR-81	Monday, the Fifth of October, 1981
	MARTIN	28-SEP-81	Monday, the Twenty-Ninth of March, 1982
1981	ALLEN	20-FEB-81	Monday, the Twenty-Fourth of August,
	TURNER	08-SEP-81	Monday, the Fifteenth of March, 1982
	JAMES	03-DEC-81	Monday, the Seventh of June, 1982
1981	WARD	22-FEB-81	Monday, the Twenty-Fourth of August,
	FORD	03-DEC-81	Monday, the Seventh of June, 1982
	SMITH	17-DEC-80	Monday, the Twenty-Second of June, 1981
	SCOTT	09-DEC-82	Monday, the Thirteenth of June, 1983
	ADAMS	12-JAN-83	Monday, the Eighteenth of July, 1983
	MILLER	23-JAN-82	Monday, the Twenty-Sixth of July, 1982

14 rows selected.

6. Para cada funcionário exiba o nome do mesmo e calcule o número de meses entre hoje e a sua data de admissão. Coloque um label na coluna MONTHS_WORKED. Ordene os resultados por número de meses empregado. Arredonde para cima o número de meses para o número inteiro mais próximo.

ENAME	MONTHS_WORKED
ADAMS	177
SCOTT	178
MILLER	188
JAMES	190
FORD	190
KING	191
MARTIN	192
TURNER	193
CLARK	196
BLAKE	197
JONES	198
WARD	199
ALLEN	199

SMITH
14 rows selected

202

Exercício 3 (continuação)

7. Crie uma consulta que produza as seguintes informações para cada funcionário: <nome do funcionário> recebe <salário> mensalmente mas deseja <salário multiplicado por 3>. Coloque um label na coluna.

Dream Salaries

```
-----  
KING earns $5,000.00 monthly but wants $15,000.00.  
BLAKE earns $2,850.00 monthly but wants $8,550.00.  
CLARK earns $2,450.00 monthly but wants $7,350.00.  
JONES earns $2,975.00 monthly but wants $8,925.00.  
MARTIN earns $1,250.00 monthly but wants $3,750.00.  
ALLEN earns $1,600.00 monthly but wants $4,800.00  
TURNER earns $1,500.00 monthly but wants $4,500.00.  
JAMES earns $950.00 monthly but wants $2,850.00.  
WARD earns $1,250.00 monthly but wants $3,750.00.  
FORD earns $3,000.00 monthly but wants $9,000.00.  
SMITH earns $800.00 monthly but wants $2,400.00.  
SCOTT earns $3,000.00 monthly but wants $9,000.00.  
ADAMS earns $1,100.00 monthly but wants $3,300.00  
MILLER earns $1,300.00 monthly but wants $3,900.00.  
14 rows selected.
```

Se você tiver tempo, complete os exercícios abaixo:

8. Crie uma consulta que exiba o nome e o salário de todos os funcionários. Formate o salário para ter 15 caracteres e apresentar o sinal US\$ à esquerda. Coloque um label na coluna SALARY.

ENAME	SALARY
-----	-----
SMITH	\$\$\$\$\$\$\$\$\$\$\$800
ALLEN	\$\$\$\$\$\$\$\$\$\$\$1600
WARD	\$\$\$\$\$\$\$\$\$\$\$1250
JONES	\$\$\$\$\$\$\$\$\$\$\$2975
MARTIN	\$\$\$\$\$\$\$\$\$\$\$1250
BLAKE	\$\$\$\$\$\$\$\$\$\$\$2850
CLARK	\$\$\$\$\$\$\$\$\$\$\$2450
SCOTT	\$\$\$\$\$\$\$\$\$\$\$3000
KING	\$\$\$\$\$\$\$\$\$\$\$5000
TURNER	\$\$\$\$\$\$\$\$\$\$\$1500
ADAMS	\$\$\$\$\$\$\$\$\$\$\$1100
JAMES	\$\$\$\$\$\$\$\$\$\$\$950
FORD	\$\$\$\$\$\$\$\$\$\$\$3000
MILLER	\$\$\$\$\$\$\$\$\$\$\$1300

14 rows selected.

Exercício 3 (continuação)

9. 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, para todos os funcionários cujo nome começa com J, A ou M. Forneça a cada coluna um label apropriado.

Name	Length
Jones	5
Martin	6
Allen	5
James	5
Adams	5
Miller	6

6 rows selected.

10. Exiba o nome, a data de admissão e o dia da semana em que o funcionário começou a trabalhar. Coloque um label na coluna DAY. Ordene os resultados por dia da semana, iniciando por Segunda.

ENAME	HIREDATE	DAY
MARTIN	28-SEP-81	MONDAY
CLARK	09-JUN-81	TUESDAY
KING	17-NOV-81	TUESDAY
TURNER	08-SEP-81	TUESDAY
SMITH	17-DEC-80	WEDNESDAY
ADAMS	12-JAN-83	WEDNESDAY
JONES	02-APR-81	THURSDAY
FORD	03-DEC-81	THURSDAY
SCOTT	09-DEC-82	THURSDAY
JAMES	03-DEC-81	THURSDAY
ALLEN	20-FEB-81	FRIDAY
BLAKE	01-MAY-81	FRIDAY
MILLER	23-JAN-82	SATURDAY
WARD	22-FEB-81	SUNDAY

14 rows selected.

Exercício 3 (continuação)

Se você quiser mais desafios, complete os exercícios abaixo:

11. Crie uma consulta que exibirá o nome do funcionário e o valor da comissão. Se o funcionário não receber comissão, coloque "No Commission". Coloque um label na coluna COMM.

```
ENAME      COMM
-----
SMITH      No Commission
ALLEN      300
WARD       500
JONES      No Commission
MARTIN     1400
BLAKE      No Commission
CLARK      No Commission
SCOTT      No Commission
KING       No Commission
TURNER     0
ADAMS      No Commission
JAMES      No Commission
FORD       No Commission
MILLER     No Commission
14 rows selected.
```

12. Crie uma consulta que exiba os nomes dos funcionários e indique as quantias de seus salários através de asteriscos. Cada asterisco representa cem dólares. Classifique os dados em ordem decrescente de salário. Coloque um label na coluna EMPLOYEE_AND_THEIR_SALARIES.

```
EMPLOYEE_AND_THEIR_SALARIES
-----
KING
*****
FORD      *****
SCOTT     *****
JONES     *****
BLAKE     *****
CLARK     *****
ALLEN     *****
TURNER    *****
MILLER    *****
MARTIN    *****
WARD      *****
ADAMS     *****
```

```
JAMES      *****  
SMITH      *****  
14 rows selected.
```

Exercício 3 (continuação)

Se você quiser mais desafios, complete o exercício a seguir:

13. Crie uma consulta que exiba o grau para todos os funcionários com base no valor da coluna JOB, conforme a tabela mostrada abaixo:

<i>JOB</i>	<i>GRADE</i>
PRESIDENT	A
MANAGER	B
ANALYST	C
SALESMAN	D
CLERK	E
None of the above	O

-----	-----
CLERK	E
SALESMAN	D
SALESMAN	D
MANAGER	B
SALESMAN	D
MANAGER	B
MANAGER	B
ANALYST	C
PRESIDENT	A
SALESMAN	D
CLERK	E
CLERK	E
ANALYST	C
CLERK	E

14 rows selected.

