



ORACLE

Academy



Database Programming with SQL

5-1

Funções de Conversão

ORACLE
Academy



Objetivos

- Esta lição abrange os seguintes objetivos:
 - Fornecer um exemplo de uma conversão explícita e implícita de tipo de dados
 - Explicar por que é importante, do ponto de vista dos negócios, uma linguagem ter recursos integrados de conversão de dados
 - Construir uma consulta SQL que aplique corretamente as funções de linha única TO_CHAR, TO_NUMBER e TO_DATE para produzir o resultado desejado

Objetivos

- Esta lição abrange os seguintes objetivos:
 - Aplicar o modelo de formato de data e/ou caractere apropriado para produzir a saída desejada
 - Explicar e aplicar YY e RR para retornar o ano correto, conforme armazenado no banco de dados

Finalidade

- Imagine ter que ler todos os livros da escola em arquivos de texto sem parágrafos e letras maiúsculas
- Seria difícil
- Felizmente, existem softwares para colocar letras maiúsculas, colorir e centralizar o texto e adicionar sublinhado, negrito e gráficos
- Nos bancos de dados, usa-se funções de conversão para alterar o formato e a exibição
- Essas funções são capazes de exibir números como moeda local, aplicar uma série de formatos a datas, mostrar um horário até os segundos e calcular a qual século uma data se refere

Tipos de Dados

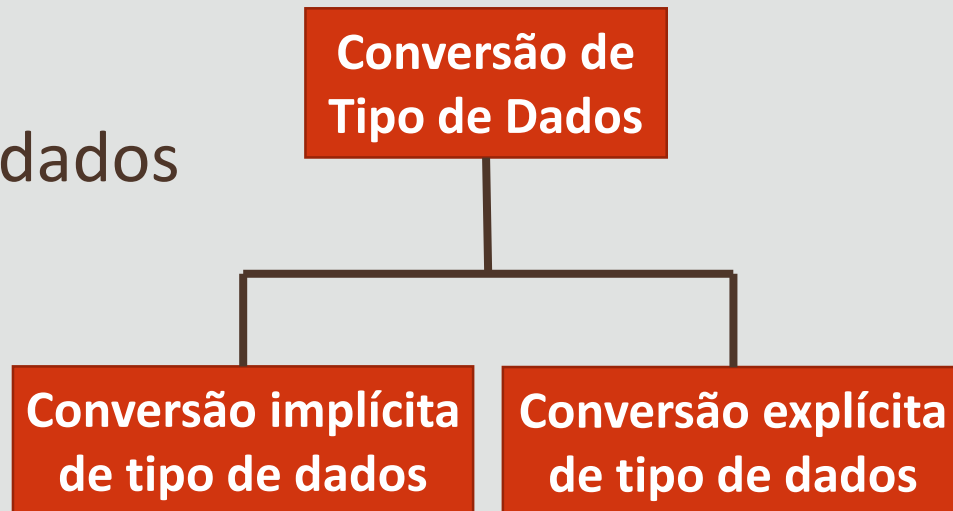
- Quando uma tabela é criada para um banco de dados, o programador de SQL deve definir o tipo de dados que será armazenado em cada campo da tabela
- Em SQL, há vários tipos diferentes de dados. Eles definem o domínio dos valores que cada coluna pode conter
- Nesta lição, você usará:
 - VARCHAR2
 - CHAR
 - NUMBER
 - DATE

Descrição dos Tipos de Dados

- VARCHAR2: usado para dados de caracteres com tamanho variável, incluindo números, traços e caracteres especiais
- CHAR: usado para texto e dados de caracteres com tamanho fixo, incluindo números, traços e caracteres especiais
- NUMBER: usado para armazenar dados numéricos com tamanho variável. Não são permitidos traços, texto ou outros dados não numéricos. Moedas são armazenadas como tipos de dados de número
- DATE: usado para valores de data e hora. Internamente, o Oracle armazena datas como números e, por padrão, as informações de DATE são exibidas no formato DD-Mon-YYYY (por exemplo, 23-Oct-2013)

Conversão de Tipo

- O Servidor Oracle é capaz de converter automaticamente dados VARCHAR2 e CHAR para os tipos NUMBER e DATE
- Ele consegue converter os dados NUMBER e DATE de volta para o tipo CHARACTER
- Isso é conhecido como conversão implícita de dados



Conversão de Tipo

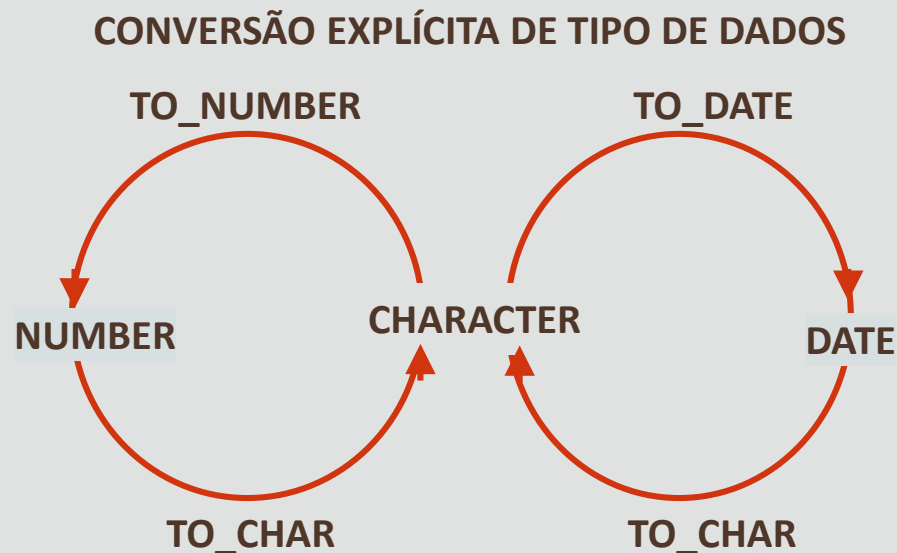
- Embora seja um recurso conveniente, é sempre melhor converter tipos de dados de maneira explícita para garantir a confiabilidade das instruções SQL

Conversões implícitas de tipo de dados

FROM	TO
VARCHAR2 or CHAR	NUMBER
VARCHAR2 or CHAR	DATE
NUMBER	VARCHAR2
DATE	VARCHAR2

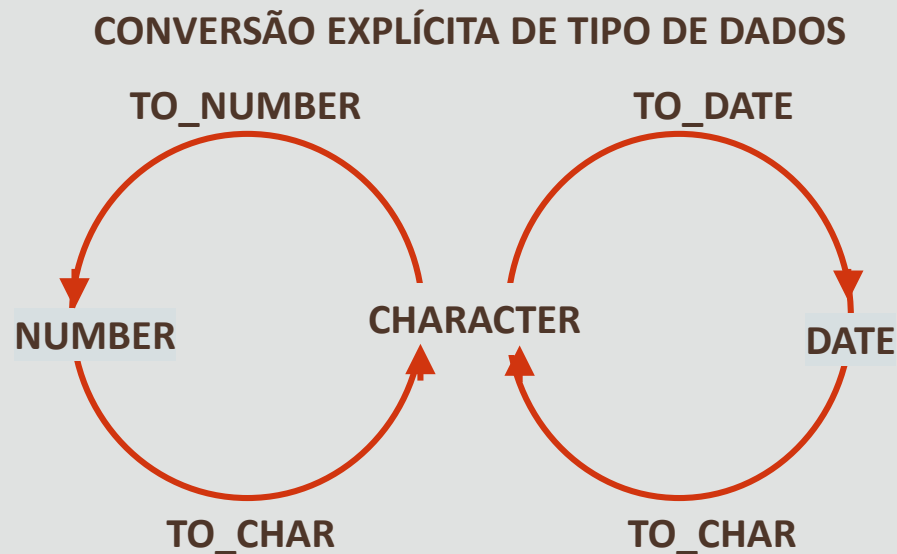
Conversão de Tipo

- As quatro funções de conversão de tipo de dados que você vai aprender são:
 - Converter o tipo de dados de data para o de caractere
 - Converter o tipo de dados de número para o de caractere



Conversão de Tipo

- As quatro funções de conversão de tipo de dados que você vai aprender são:
 - Converter o tipo de dados de caractere para o de número
 - Converter o tipo de dados de caractere para o de data



Conversão de Data para Dados de Caracteres

- Você pode muitas vezes querer converter uma data do formato padrão (DD-Mon-YYYY) para outro formato especificado
- A função para conseguir fazer isso é:

```
TO_CHAR (date column name, 'format model you specify')
```

- O 'modelo de formato' deve estar entre aspas simples e faz distinção entre letras maiúsculas e minúsculas
- Separe com uma vírgula o valor de data do modelo de formato
- Pode ser incluído qualquer elemento de formato de data válido

Conversão de Data para Dados de Caracteres

- Use sp para escrever o número por extenso
- Use th para fazer aparecer um número ordinal (1st, 2nd, 3rd etc.)
- Use um elemento fm para remover espaços em branco ou zeros à esquerda da saída

Conversão de Data para Dados de Caracteres

YYYY	Ano completo em números
YEAR	Ano escrito por extenso
MM	Valor de dois dígitos para o mês
MONTH	Nome completo do mês
MON	Abreviação com três letras do mês
DY	Abreviação com três letras do dia da semana
DAY	Nome completo do dia da semana
DD	Dia numérico do mês
DDspth	FOURTEENTH
Ddspth	Fourteenth
ddspth	fourteenth
DDD ou DD ou D	Dia do ano, mês ou semana
HH24:MI:SS AM	15:45:32 PM
DD "of" MONTH	12 de Outubro

- As tabelas mostram os diferentes modelos de formatos que podem ser usados
- Ao especificar elementos de hora, observe que as horas (HH), os minutos (MI), os segundos (SS) e o período do dia (manhã (AM); tarde ou noite (PM)) também podem ser formatados

Conversão de Data para Dados de Caracteres

- Exemplos de saídas usando diferentes modelos de formatos:

Examples:	Output
<pre>SELECT TO_CHAR(hire_date, 'Month dd, YYYY') FROM employees;</pre>	<pre>... June 07, 1994 ...</pre>
<pre>SELECT TO_CHAR(hire_date, 'fmMonth dd, YYYY') FROM employees;</pre>	<pre>... June 7, 1994 ...</pre>
<pre>SELECT TO_CHAR(hire_date, 'fmMonth ddth, YYYY') FROM employees;</pre>	<pre>June 7th, 1994 January 3rd, 1990 ...</pre>

Conversão de Data para Dados de Caracteres

- Exemplos de saídas usando diferentes modelos de formatos:

Exemplos:	Saída
<code>SELECT TO_CHAR(hire_date, 'fmDay ddth Mon, YYYY') FROM employees;</code>	Tuesday 7th Jun, 1994
<code>SELECT TO_CHAR(hire_date, 'fmDay ddthsp Mon, YYYY')</code> <code>FROM employees;</code>	Tuesday, seventh Jun, 1994
<code>SELECT TO_CHAR(hire_date, 'fmDay, ddthsp "of" Month, Year')</code> <code>FROM employees;</code>	Tuesday, seventh of June, Nineteen Ninety-Four

Conversão de Data para Dados de Caracteres

- Exemplos de saídas usando diferentes modelos de formatos para hora:

Exemplos:	Saída
SELECT TO_CHAR(SYSDATE, 'hh:mm') FROM dual;	02:07
SELECT TO_CHAR(SYSDATE, 'hh:mm pm') FROM dual;	02:07 am
SELECT TO_CHAR(SYSDATE, 'hh:mm:ss pm') FROM dual;	02:07:23 am

Conversão de Número para Dados de Caracteres (VARCHAR2)

- Os números armazenados no banco de dados não têm formatação
- Isso significa que não têm sinais/símbolos de moeda, vírgulas, decimais ou outra formatação
- Para adicionar formatação, primeiro você precisa converter o número para o formato de caractere

```
TO_CHAR(number, 'format model')
```

- A função SQL que você deve usar para converter um número para o formato de caractere desejado é:

Conversão de Número para Dados de Caracteres (VARCHAR2)

- A tabela ilustra alguns dos elementos de formato disponíveis para serem usados com funções TO_CHAR

```
SELECT TO_CHAR(salary,  
 '$99,999') AS "Salary"  
FROM employees;
```

Salary
\$24,000
\$17,000

ELEMENTO	DESCRIÇÃO	EXEMPLO	RESULTADO
9	Posição numérica (a quantidade de nove determina a largura)	999999	1234
0	Exibir zeros à esquerda	099999	001234
\$	Cifrão 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	Colocar números negativos entre parênteses	999999PR	<1234>
EEEE	Notação científica (deve ter quatro Es)	99.999EEEE	1,23E+03
V	Multiplicar por 10 n vezes (n = número de nove após o V)	9999V99	9999V99
B	Exibir valores zero como um espaço em branco, e não 0	B9999.99	1234.00

Conversão de Número para Dados de Caracteres (VARCHAR2)

- Você consegue identificar os modelos de formato usados para produzir a saída a seguir?
 - \$3000.00
 - 4,500
 - 9,000.00
 - 0004422

ELEMENTO	DESCRIÇÃO	EXEMPLO	RESULTADO
9	Posição numérica (a quantidade de noves determina a largura)	999999	1234
0	Exibir zeros à esquerda	099999	001234
\$	Cifrão 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	Colocar números negativos entre parênteses	999999PR	<1234>
EEEE	Notação científica (deve ter quatro Es)	99.999EEEE	1,23E+03
V	Multiplicar por 10 n vezes (n = número de noves após o V)	9999V99	9999V99
B	Exibir valores zero como um espaço em branco, e não 0	B9999.99	1234.00

Conversão de Número para Dados de Caracteres (VARCHAR2)

- Respostas:

SQL:	Saída
SELECT TO_CHAR(3000, '\$99999.99') FROM dual;	\$3000.00
SELECT TO_CHAR(4500, '99,999') FROM dual;	4,500
SELECT TO_CHAR(9000, '99,999.99') FROM dual;	9,000.00
SELECT TO_CHAR(4422, '0009999') FROM dual;	0004422

Conversão de Caracteres para Número

- Você pode muitas vezes querer converter uma string de caracteres para um número. A função para essa conversão é:

```
TO_NUMBER(character string, 'format model')
```

- O modelo de formato é opcional, mas deve ser incluído se a string de caracteres sendo convertida tiver caracteres que não sejam números
- Não é possível fazer cálculos precisos com dados de caracteres

```
SELECT TO_NUMBER('5,320', '9,999')  
AS "Number"  
FROM dual;
```

Number
5320

Conversão de Caracteres para Número

- A coluna de bônus inclui dados que contêm quatro caracteres. O modelo de formato especifica três caracteres. Portanto, é retornado um erro

```
SELECT last_name, TO_NUMBER(bonus,  
'999')  
FROM employees  
WHERE department_id = 80;
```



ORA-01722: invalid number

```
SELECT last_name, TO_NUMBER(bonus,  
'9999')  
  AS "Bonus"  
FROM employees  
WHERE department_id = 80;
```

LAST_NAME	Bonus
Zlotkey	1500
Abel	1700
Taylor	1250

Conversão de Caracteres para Data

- Para converter uma string de caracteres para um formato de data, use:

```
TO_DATE('character string', 'format model')
```

- Essa conversão seleciona uma string de caracteres com um valor que não seja de data, como "November 3, 2001", e a torna um valor de data
- O modelo de formato informa o servidor qual é a "aparência" da string de caracteres:

```
TO_DATE('November 3, 2001', 'Month dd, yyyy')
```

– retornará 03-Nov-2001

Conversão de Caracteres para Data

- Ao fazer uma conversão de caracteres para data, o modificador fx (formato exato) especifica a correspondência exata do argumento de caracteres e o modelo de formato de data
- No exemplo a seguir, observe que "May10" não tem um espaço entre "May" e "10"
- O modelo de formato fx corresponde ao argumento de caracteres, já que também não há um espaço entre "Mon" e "DD"

```
SELECT TO_DATE('May10,1989', 'fxMonDD,YYYY')  
AS "Convert"  
FROM DUAL;
```

CONVERT
10-May-1989

Regras do Modificador fx

- As regras do modificador fx são:
 - A pontuação e o texto entre aspas no argumento de caracteres devem ser idênticos às partes correspondentes do modelo de formato (exceto letras maiúsculas e minúsculas)
 - O argumento de caracteres não pode ter espaços em branco extras
 - Sem o fx, o Servidor Oracle ignora esses espaços.
 - Os dados numéricos no argumento de caracteres deve ter a mesma quantidade de dígitos do elemento correspondente no modelo de formato
 - Sem o fx, os números no argumento de caracteres podem omitir zeros à esquerda

Regras do Modificador fx

Examples:	Output
SELECT TO_DATE('Sep 07, 1965', 'fxMon dd, YYYY') AS "Date" FROM dual;	07-Sep-1965
SELECT TO_DATE('July312004', 'fxMonthDDYYYY') AS "Date" FROM DUAL;	31-Jul-2004
SELECT TO_DATE('June 19, 1990','fxMonth dd, YYYY') AS "Date" FROM DUAL;	19-Jun-1990

Formatos de Data RR e YY

- Todos os dados de data devem agora ser armazenados usando os quatros dígitos dos anos (YYYY)
- No entanto, é possível que alguns bancos de dados legados ainda usem o formato com dois dígitos (YY)
- Não faz tanto tempo que o século mudou de 1900 para 2000
- Com essa mudança, ficou muito confuso saber se uma data gravada como 02-Jan-98 seria interpretada como January 2, 1998 ou January 2, 2098

Formatos de Data RR e YY

- Se os dados que estão sendo convertidos de caracteres para data contiverem somente um ano com dois dígitos, o Oracle tem uma maneira de interpretar essa data no século correto
- Por exemplo: '27/OCT/95'

```
SELECT TO_DATE('27-Oct-95', 'DD-Mon-YY')  
       AS "Date"  
FROM dual;
```

Date
27-Oct-2095

- O ano com dois dígitos foi interpretado como 2095. Isso pode não ser o que se queria

Formatos de Data RR e YY

- Se YY for usado no modelo de formato, supõe-se que o ano esteja no século correto
- Se o ano com dois dígitos não estiver no século atual, usamos RR

```
SELECT TO_DATE('27-Oct-95', 'DD-Mon-RR')  
AS "Date"  
FROM dual;
```

Date
27-Oct-1995

- O ano com dois dígitos agora é interpretado como 1995

Algumas Regras Simples

- Se o formato de data for especificado com RR, haverá duas possibilidades de valor de retorno, dependendo do ano atual
- Se o ano atual estiver entre 00 e 49:
 - Datas de 0 a 49:
A data estará no século atual
 - Datas de 50 a 99:
A data estará no século passado

		Se o ano com dois dígitos especificado estiver entre:	
		0 e 49	50 e 99
Se dois dígitos do ano atual especificado estiverem entre:	0 e 49	A data de retorno estará no século atual	A data de retorno estará no século anterior ao atual
	50 e 99	A data de retorno estará no século posterior ao atual	A data de retorno estará no século atual

Algumas Regras Simples

- Se o ano atual estiver entre 50 e 99:
 - Datas de 0 a 49: A data estará no próximo século
 - Datas de 50 a 99: A data estará no século atual

		Se o ano com dois dígitos especificado estiver entre:	
		0 e 49	50 e 99
Se dois dígitos do ano atual especificado estiverem entre:	0 e 49	A data de retorno estará no século atual	A data de retorno estará no século anterior ao atual
	50 e 99	A data de retorno estará no século posterior ao atual	A data de retorno estará no século atual

Algumas Regras Simples

- A tabela abaixo oferecem alguns exemplos de como YY e RR são interpretados, dependendo do ano atual

Current Year	Specified Date	RR Format	YY Format
1995	27-Oct-95	1995	1995
1995	27-Oct-17	2017	1917
2015	27-Oct-17	2017	2017
2015	27-Oct-95	1995	2095

Algumas Regras Simples

- Quando consulto meu banco de dados de funcionários usando a instrução a seguir, ela retorna todas as linhas na tabela
- Sei que apenas alguns funcionários foram contratados antes de 1990

```
SELECT last_name, TO_CHAR(hire_date, 'DD-Mon-YY')  
FROM employees  
WHERE hire_date < TO_DATE('01-Jan-90', 'DD-Mon-YY');
```

- Como o modelo de formato na cláusula WHERE usa YY e o ano atual é 2015, a consulta retorna linhas com data_contratação anterior a 2090

Terminologia

- Estes são os principais termos usados nesta lição:
 - CHAR
 - DATE
 - Formato de data DD
 - Função de conversão
 - fm
 - NUMBER

Terminologia

- Estes são os principais termos usados nesta lição:
 - Formato de data RR
 - TO_CHAR
 - TO_DATE
 - TO_NUMBER
 - VARCHAR2
 - Modificador fx

Resumo

- Nesta lição, você deverá ter aprendido a:
 - Fornecer um exemplo de uma conversão explícita e implícita de tipo de dados
 - Explicar por que é importante, do ponto de vista dos negócios, uma linguagem ter recursos integrados de conversão de dados
 - Construir uma consulta SQL que aplique corretamente as funções de linha única TO_CHAR, TO_NUMBER e TO_DATE para produzir o resultado desejado

Resumo

- Nesta lição, você deverá ter aprendido a:
 - Aplicar o modelo de formato de data e/ou caractere apropriado para produzir a saída desejada
 - Explicar e aplicar YY e RR para retornar o ano correto, conforme armazenado no banco de dados



ORACLE

Academy

