Conversion Functions

What Will I Learn?

In this lesson, you will learn to:

- Provide an example of an explicit data-type conversion and an implicit data-type conversion
- Explain why it is important, from a business perspective, for a language to have built-in data-conversion capabilities
- Construct a SQL query that correctly applies TO_CHAR, TO_NUMBER and TO_DATE single-row functions to produce a desired result
- Apply the appropriate date and/or character format model to produce a desired output
- Explain and apply the use of YYYY and RRRR to return the correct year as stored in the database

Why Learn It?

Imagine having to read all your school books in text files with no paragraphs and no capitalization. It would be difficult to read. Fortunately, there are software programs available to capitalize and color text, underline, bold, center and add graphics.

For databases, format and display changes are done using conversion functions. These functions are able to display numbers as local currency, format dates in a variety of formats, display time to the second, and keep track of what century a date refers to.

Tell Me / Show Me

When a table is created for a database, the SQL programmer must define what kind of data will be stored in each field of the table. In SQL, there are several different data types. These data types define the domain of values that each column can contain. For this lesson, you will use:

VARCHAR2

CHAR

NUMBER

DATE

Tell Me / Show Me

Data Types

VARCHAR2: Used for character data of variable length, including numbers, special characters and dashes.

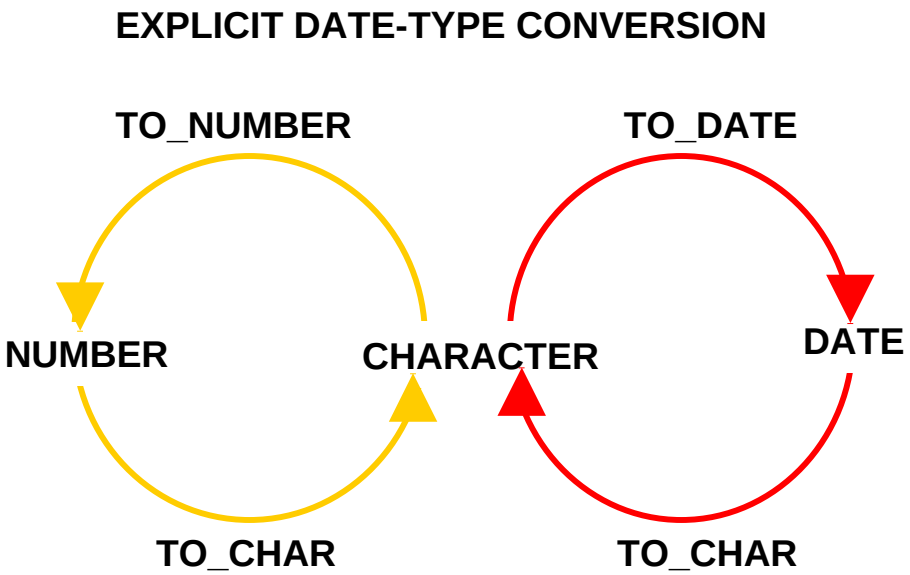
CHAR: Used for text and character data of fixed length, including numbers, dashes and special characters.

NUMBER: Used to store variable-length numeric data. No dashes, text, or other nonnumeric data are allowed. Currency is stored as a number data type.

DATE: Used for date and time values. Internally, Oracle stores dates as numbers and by default DATE information is displayed as DD-MON-YY (for example, 19-JUN-04).

Tell Me / Show Me

The Oracle Server can internally convert VARCHAR2 and CHAR data to NUMBER and DATE data types. It can convert NUMBER and DATE data back to CHARACTER datatype. Although this is a convenient feature, it is always best to explicitly make date-type conversions to ensure reliability in SQL statements.



The four data-type conversion functions you will learn are:

- To convert date data type to character data type
- To convert number data type to character data type
- To convert character data type to number data type
- To convert character data type to date data types

Tell Me / Show Me

Date Conversion to Character Data

It is often desirable to convert dates stored in a database in the default DD-MON-YY format to another format specified by you. The function to accomplish this task is:

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

- The ‘format model’ must be enclosed in single quotation marks and is case-sensitive.
- Separate the date value from the ‘format model with a comma.
- Any valid date format element can be included.
- Use an fm element to remove padded blanks or remove leading zeroes from the output.
- Use sp to spell out a number.
- Use th to have the number appear as an ordinal. (1st, 2nd, 3rd and so on)
- Use double quotation marks to add character strings to format models.

Tell Me / Show Me

Date Conversion to Character Data (continued)

The tables show the different format models that can be used. When specifying time elements, note that hours (HH), minutes (MI), seconds (SS), and AM or PM can also be formatted.

YYYY	Full year in numbers
YEAR	Year spelled out
MM	Two-digit value for month
MONTH	Full name of the month
MON	Three-letter abbreviation of the month
DY	Three-letter abbreviation of the day of the week
DAY	Full name of the day of the week
DD	Numeric day of the month

HH24:MI:SS AM	15:45:32 PM
DD “of” MONTH	12 of October

DDspth	FOURTEENTH
Ddspth	Fourteenth
ddspth	fourteenth
DDD or DD or D	Day of year, month or week

Tell Me / Show Me

Date Conversion to Character Data (continued)

For example, the following query returns May 14, 2004. If an event date were 04-MAY-04, the format model using fm would have returned May 4, 2004, suppressing the leading zero.

```
SELECT TO_CHAR (event_date, 'fmMonth dd, RRRR')
FROM d_events;
```

What will be the output of the following query?

```
SELECT id, TO_CHAR(event_date, 'MONTH DD, YYYY')
FROM d_events;
```

Tell Me / Show Me

Date and Time Format Models

The following tables show variations of the format models that can be used with dates and time. Can you identify the format models used to produce today's date as the following output?

August 6th, 2007

August 06, 2007

AUG 6, 2007

August 6th, Friday, Two Thousand Seven

YYYY	Full year in numbers
YEAR	Year spelled out
MM	Two-digit value for month
MONTH	Full name of the month
MON	Three-letter abbreviation of the month
DY	Three-letter abbreviation of the day of the week
DAY	Full name of the day of the week
DD	Numeric day of the month

HH24:MI:SS AM	15:45:32 PM
DD “of” MONTH	12 of October

DDspth	FOURTEENTH
Ddspth	Fourteenth
ddspth	fourteenth
DDD or DD or D	Day of year, month or week

Tell Me / Show Me

Number Conversion to Character Data (VARCHAR2)

Numbers stored in the database have no formatting. This means that there are no currency signs/symbols no commas, no decimals or other formatting. To add formatting, you first need to convert the number to a character format. This conversion is especially useful with concatenation.

The SQL function that you use to convert columns of number data to a desired format is:

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

Tell Me / Show Me

Number Conversion to Character Data (VARCHAR2) (continued)

The table illustrates some of the format elements available to use with TO_CHAR functions.

```
SELECT TO_CHAR(cost,
'$99,999') COST
FROM d_events;
```

COST
$8,000
$10,000

ELEMENT	DESCRIPTION	EXAMPLE	RESULT
9	Numeric position (# of 9's determine width)	999999	1234
0	Display leading zeros	099999	001234
$	Floating dollar sign	$999999	$1234
L	Floating local currency symbol	L999999	FF1234
.	Decimal point in position specified	999999.99	1234.00
,	Comma in position specified	999,999	1,234
MI	Minus signs to right (negative values)	999999MI	1234-
PR	Parenthesize negative numbers	999999PR	<1234>
EEEE	Scientific notation (must have four EEEE)	99.999EEEE	1,23E+03
V	Multiply by 10 n times (n= number of 9's after V)	9999V99	9999V99
B	Display zero values as blank, not 0	B9999.99	1234.00

Tell Me / Show Me

Number Conversion to Character Data (VARCHAR2) (continued)

Can you identify the format models used to produce the following output?

$3000.00

4,500

9,000.00

0004422

ELEMENT	DESCRIPTION	EXAMPLE	RESULT
9	Numeric position (# of 9's determine width)	999999	1234
0	Display leading zeros	099999	001234
$	Floating dollar sign	$999999	$1234
L	Floating local currency symbol	L999999	FF1234
.	Decimal point in position specified	999999.99	1234.00
,	Comma in position specified	999,999	1,234
MI	Minus signs to right (negative values)	999999MI	1234-
PR	Parenthesize negative numbers	999999PR	<1234>
EEEE	Scientific notation (must have four EEEE)	99.999EEEE	1,23E+03
V	Multiply by 10 n times (n= number of 9's after V)	9999V99	9999V99
B	Display zero values as blank, not 0	B9999.99	1234.00

Tell Me / Show Me

Character Conversion to Number

It is often desirable to convert a character string to a number. The function for this conversion is:

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

This converts a nonnumeric value such as "450" to a number, without the single quotes. The single quotes are characters. The "450" was stored in the database as character data, and the following query converts it to a number so that arithmetic operations can be performed. You cannot perform calculations with character data.

```
SELECT TO_NUMBER('450‘) AS "Number Change“
FROM DUAL;
```

Number Change
450

Tell Me / Show Me

Character Conversion to Number (continued)

```
SELECT TO_NUMBER('450', '9999') + 10 AS "Number
Change" FROM DUAL;
```

Number Change
460

SQL*Plus displays a string of hash signs (#) in place of a whole number whose digits exceed the number of digits provided in the format model and rounds numbers to the decimal place provided in the format model.

Oracle Application Express will return an Oracle Error – Invalid Number, if the format model does not match the actual number returned by the database.

Tell Me / Show Me

Character Conversion to Date

To convert a character string to a date format, use:

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

This conversion takes a nondate value character string such as "November 3, 2001" and converts it to a date value. The format model tells the server what the character string "looks like":

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

will return 03-NOV-01

Tell Me / Show Me

Character Conversion to Date (continued)

When making a character-to-date conversion, the fx (format exact) modifier specifies exact matching for the character argument and the date format model.

In the following example, note that "May10" has no space between "May" and "10." The fx format model matches the character argument as it also has no space between "Mon" and "DD."

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

CONVERT
10-MAY-89

Tell Me / Show Me

fx Modifier Rules

The fx modifier rules are:

- Punctuation and quoted text in the character argument must match the corresponding parts of the format model exactly (except for case).
- The character argument cannot have extra blanks. Without fx, the Oracle Server ignores extra blanks.
- Numeric data in the character argument must have the same number of digits as the corresponding element in the format model. Without fx, numbers in the character argument can omit leading zeros.

Tell Me / Show Me

RR Date Format and YY Date Format

It hasn't been that long since the century changed from 1900 to 2000. Along with this change came considerable confusion as to whether a date written as 02-JAN-00 would be interpreted as January 2, 1900 or January 2, 2000. Fortunately, Oracle has a way to keep these dates stored and retrievable with the correct century.

Current Year	Specified Date	RR Format	YY Format
1995	27-OCT-95	1995	1995
1995	27-OCT-17	2017	1917
2001	27-OCT-17	2017	2017
2001	27-OCT-95	1995	2095

		If the specified two-digit year is:	
		0-49	50-99
If two digits of the current year are:	0-49	The return date is in the current century	The return date is in the century before the current one
	50-99	The return date is in the century after the current one	The return date is in the current century

Tell Me / Show Me

A Few Simple Rules

If the date format is specified with the YY or YYYY format, the return value will be in the same century as the current century. So, if the year is 1995 and you use the YY or YYYY format, all is well and the dates will be in the 1900s. However, if the current year is 2004 and you use the YY or YYYY format for a date such as 1989, you will get 2089! Maybe not what you intended.

Current Year	Specified Date	RR Format	YY Format
1995	27-OCT-95	1995	1995
1995	27-OCT-17	2017	1917
2001	27-OCT-17	2017	2017
2001	27-OCT-95	1995	2095

Tell Me / Show Me

A Few Simple Rules (continued)

If the date format is specified with the RR or RRRR format, the return value has two possibilities.

If the current year is between 00-49:

- Dates from 0-49: The date will be in the current century
- Dates from 50-99: The date will be in the last century

If the current year is between 50-99:

- Dates from 0-49: The date will be in next century
- Dates from 50-99: The date will be in current century

		If the specified two-digit year is:	
		0-49	50-99
If two digits of the current year are:	0-49	The return date is in the current century	The return date is in the century before the current one
	50-99	The return date is in the century after the current one	The return date is in the current century

Tell Me / Show Me

A Few Simple Rules (continued)

When I query my employee database using the following statement, it returns every row in the table. I know there are only a few employees who were hired before 1990. Did I make a mistake?

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

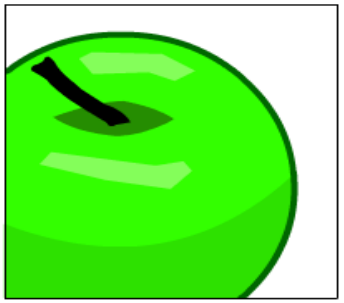
Current Year	Specified Date	RR Format	YY Format
1995	27-OCT-95	1995	1995
1995	27-OCT-17	2017	1917
2001	27-OCT-17	2017	2017
2001	27-OCT-95	1995	2095

		If the specified two-digit year is:	
		0-49	50-99
If two digits of the current year are:	0-49	The return date is in the current century	The return date is in the century before the current one
	50-99	The return date is in the century after the current one	The return date is in the current century

Tell Me / Show Me

Terminology

Key terms used in this lesson include:



CHAR

DATE

DD date format

Conversion function

fm

NUMBER

RR date format

TO_CHAR

TO_DATE

TO_NUMBER

VARCHAR2

Summary

In this lesson, you have learned to:

- Provide an example of an explicit data-type conversion and an implicit data-type conversion
- Explain why it is important, from a business perspective, for a language to have built-in data-conversion capabilities
- Construct a SQL query that correctly applies TO_CHAR, TO_NUMBER and TO_DATE single-row functions to produce a desired result
- Apply the appropriate date and/or character format model to produce a desired output
- Explain and apply the use of YYYY and RRRR to return the correct year as stored in the database

Summary

Practice Guide

The link for the lesson practice guide can be found in the course resources in Section 0.