ORACLE

Academy

Database Programming with SQL

5-1

Conversion Functions

ORACLE
Academy

Objectives

- This lesson covers the following objectives:
  - 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

Objectives

- This lesson covers the following objectives:
  - Apply the appropriate date and/or character format model to produce a desired output
  - Explain and apply the use of YY and RR to return the correct year as stored in the database

Purpose

- 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

Data Types

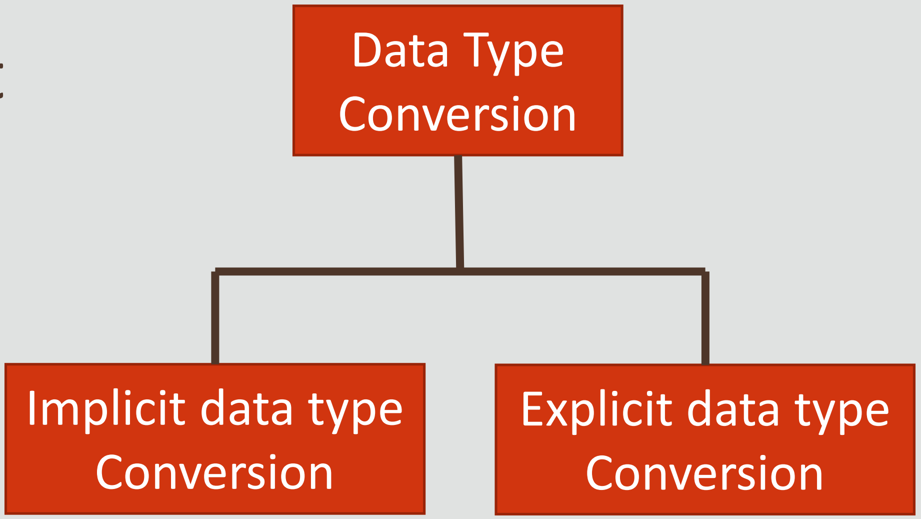
- When a table is created for a database, the SQL programmer must define what kind of data will be stored in each field of the table
- In SQL, there are several different data types. These data types define the domain of values that each column can contain
- For this lesson, you will use:
  - VARCHAR2
  - CHAR
  - NUMBER
  - DATE

Data Types Described

- VARCHAR2: Used for character data of variable length, including numbers, dashes, and special characters
- 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-YYYY (for example, 23-Oct-2013)

Type Conversion

- The Oracle Server can automatically convert VARCHAR2 and CHAR data to NUMBER and DATE data types
- It can convert NUMBER and DATE data back to CHARACTER data type
- This is known as implicit data conversion



Type Conversion

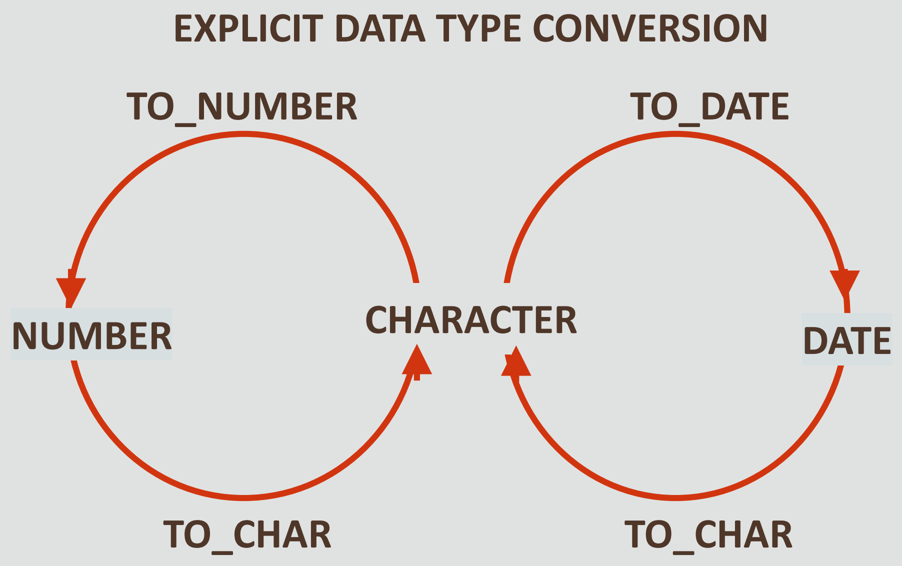
- Although this is a convenient feature, it is always best to explicitly make data type conversions to ensure reliability in SQL statements

Implicit data type conversions

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

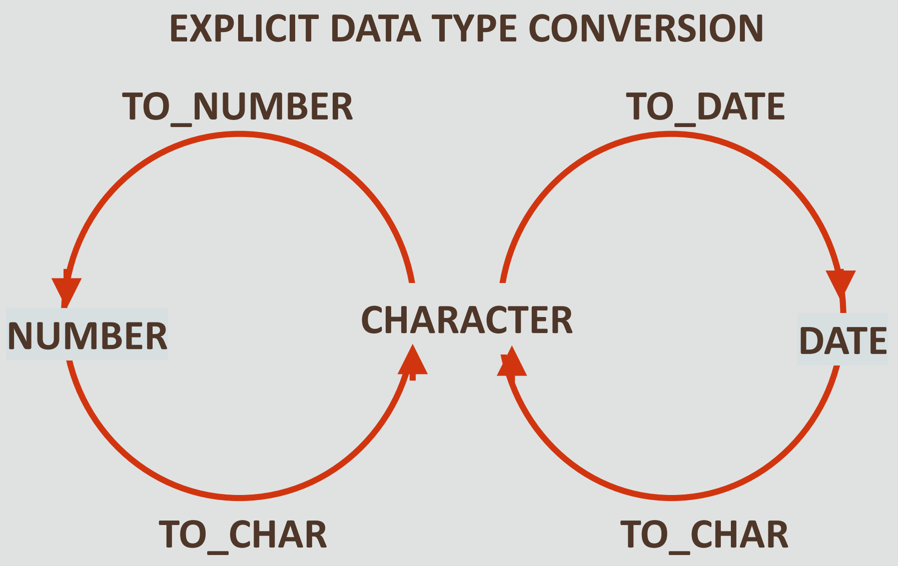
Type Conversion

- 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



Type Conversion

- The four data-type conversion functions you will learn are:
  - To convert character data type to number data type
  - To convert character data type to date data types



Date Conversion to Character Data

- It is often desirable to convert a date from its default DD-Mon-YYYY 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

Date Conversion to Character Data

- Use sp to spell out a number
- Use th to have the number appear as an ordinal
  - (1st, 2nd, 3rd, and so on)
- Use an fm element to remove padded blanks or remove leading zeroes from the output

Date Conversion to Character Data

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
DDspth	FOURTEENTH
Ddspth	Fourteenth
ddspth	fourteenth
DDD or DD or D	Day of year, month or week
HH24:MI:SS AM	15:45:32 PM
DD "of" MONTH	12 of October

- 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

Date Conversion to Character Data

- Examples of output using different format models:

Examples:	**Output**

SELECT TO_CHAR(hire_date, 'Month dd, YYYY')
FROM employees;
	…
	June 07, 1994
	…

SELECT TO_CHAR(hire_date, 'fmMonth dd, YYYY')
FROM employees;
	…
	June 7, 1994
	…

SELECT TO_CHAR(hire_date, 'fmMonth ddth, YYYY')
FROM employees;
	June 7th, 1994
	January 3rd, 1990
	…

Date Conversion to Character Data

- Examples of output using different format models:

Examples:	Output
SELECT TO_CHAR(hire_date, 'fmDay ddth Mon, YYYY') FROM employees;	Tuesday 7th Jun, 1994
SELECT TO_CHAR(hire_date, 'fmDay ddthsp Mon, YYYY') FROM employees;	Tuesday, seventh Jun, 1994
SELECT TO_CHAR(hire_date, 'fmDay, ddthsp "of" Month, Year') FROM employees;	Tuesday, seventh of June, Nineteen Ninety-Four

Date Conversion to Character Data

- Examples of output using different format models for time:

Examples:	**Output**

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

Number Conversion to Character Data (VARCHAR2)

- Numbers stored in the database have no formatting
- This means that they have no currency signs/symbols, commas, decimals, or other formatting
- To add formatting, you first need to convert the number to a character format

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

- The SQL function that you use to convert a number to a desired character format is:

Number Conversion to Character Data (VARCHAR2)

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

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

Salary
$24,000
$17,000

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

Number Conversion to Character Data (VARCHAR2)

- 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

ORACLE Academy

Number Conversion to Character Data (VARCHAR2)

- Answers:

SQL:	**Output**

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

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')
```

- The format model is optional, but should be included if the character string being converted contains any characters other than numbers
- You cannot reliably perform calculations with character data

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

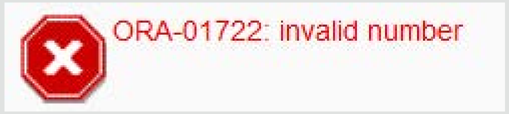
Number
5320

ORACLE Academy

Character Conversion to Number

- The bonus column includes data which contains 4 characters, the format model specifies 3 characters, so an error is returned

```
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

Character Conversion to Date

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

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

- This conversion takes a non-date 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, yyyy')
```

  - will return 03-Nov-2001

Character Conversion to Date

- 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,YYYY') AS "Convert"
FROM DUAL;
```

CONVERT
10-May-1989

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

fx Modifier Rules

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

RR Date Format and YY Date Forma

- All date data should now be stored using four-digit years (YYYY)
- Some legacy databases however may still use the two-digit (YY) format
- It has not 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-98 would be interpreted as January 2, 1998 or January 2, 2098

RR Date Format and YY Date Format

- If the data being converted from character data to date data contains only a two-digit year, Oracle has a way of interpreting these dates in the correct century
- For example: '27-Oct-95'

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

Date
27-Oct-2095

- The two-digit year is interpreted as 2095, this may not be what was intended

ORACLE Academy

RR Date Format and YY Date Format

- If YY is used in the format model, the year is assumed to be in the current century
- If the two-digit year is not in the current century, we use RR

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

Date
27-Oct-1995

- The two-digit year is now interpreted as 1995

A Few Simple Rules

- If the date format is specified with the RR format, the return value has two possibilities, depending on the current year
- 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 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

A Few Simple Rules

- 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

A Few Simple Rules

- The table below gives some examples of how YY and RR are interpreted, depending on the current year

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

A Few Simple Rules

- 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

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

- As the format model in the WHERE clause uses YY, and the current year is 2017, the query returns rows with a hire_date less than 2090

ORACLE
Academy

Terminology

- Key terms used in this lesson included:
  - CHAR
  - DATE
  - DD date format
  - Conversion function
  - fm
  - NUMBER

Terminology

- Key terms used in this lesson included:
  - RR date format
  - TO_CHAR
  - TO_DATE
  - TO_NUMBER
  - VARCHAR2
  - Fx Modifier

Summary

- In this lesson, you should have learned how 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

Summary

- In this lesson, you should have learned how to:
  - Apply the appropriate date and/or character format model to produce a desired output
  - Explain and apply the use of YY and RR to return the correct year as stored in the database

ORACLE
Academy