



ORACLE

Academy



Database Programming with SQL

5-1

Funciones de Conversión

ORACLE
Academy



Objetivos

- En esta lección se abordan los siguientes objetivos:
 - Proporcionar un ejemplo de conversión de tipo de dato implícita y explícita y una conversión de tipo de dato implícita
 - Explicar el motivo de la importancia, desde una perspectiva de negocio, de que un lenguaje tenga capacidades de conversión de datos incorporadas
 - Crear una consulta SQL que aplique correctamente las funciones de una sola fila TO_CHAR, TO_NUMBER y TO_DATE para obtener el resultado deseado

Objetivos

- En esta lección se abordan los siguientes objetivos:
 - Aplicar el modelo de formato de fecha y/o caracteres adecuado para producir una salida deseada
 - Explicar y aplicar el uso de YY y RR para devolver el año correcto como se almacenó en la base de datos

Objetivo

- Imagine que tuviera que leer todos los libros de texto en archivos de texto sin párrafos y sin mayúsculas
- Sería difícil leerlos
- Afortunadamente, existen programas de software disponibles que permiten poner en mayúsculas y aplicar color al texto, subrayarlos, ponerlos en negrita, centrarlo y agregar gráficos
- En el caso de bases de datos, los cambios de formato y visualización se realizan utilizando funciones de conversión
- Estas funciones permiten mostrar los números en la moneda local, aplicar una gran variedad de formatos a las fechas, mostrar la hora incluyendo hasta los segundos y hacer un seguimiento del siglo al que hace referencia una fecha



Tipos de Dato

- Cuando se crea una tabla para una base de datos, el programador SQL debe definir qué tipo de dato se almacenará en cada uno de los campos de la tabla
- En SQL, hay diferentes tipos de dato. Estos tipos de dato definen el dominio de valores que puede incluir cada columna
- Para esta lección, utilizará:
 - VARCHAR2
 - CHAR
 - NUMBER
 - DATE

Descripción de Tipos de Dato

- VARCHAR2: se utiliza para datos de caracteres de longitud variable, incluidos los números, los guiones y los caracteres especiales
- CHAR: se utiliza para datos de texto y de caracteres de longitud fija, incluidos los números, los guiones y los caracteres especiales

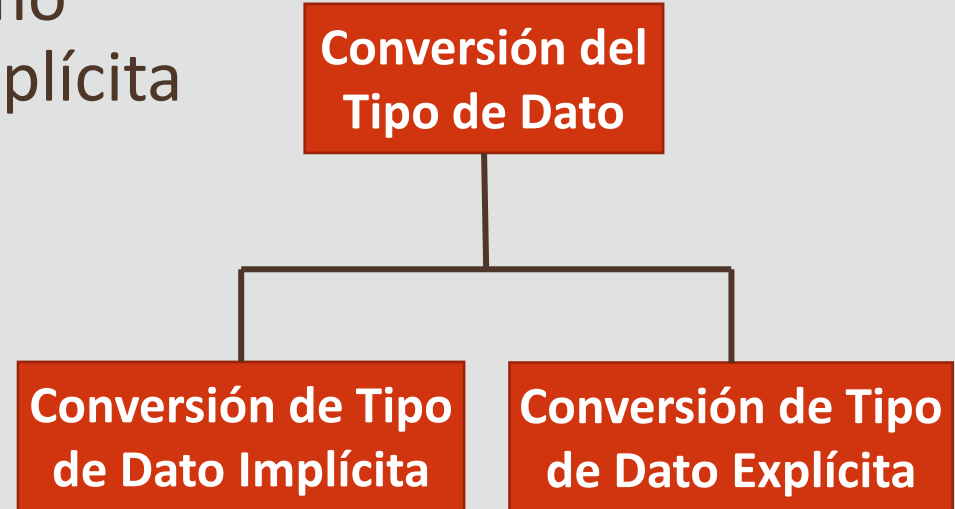
Descripción de Tipos de Dato

- **NUMBER:** se utiliza para almacenar datos numéricos de longitud variable. No se permiten guiones, texto u otros datos no numéricos. La moneda se almacena como tipo de dato numérico
- **DATE:** se utiliza para valores de fecha y hora. Internamente, Oracle almacena las fechas como números y, por defecto, se muestra información de DATE como DD-Mes-YYYY (por ejemplo, 23-Oct-2013)



Conversión de Tipo

- Oracle Server puede convertir automáticamente datos VARCHAR2 y CHAR en tipos de dato NUMBER y DATE
- Puede convertir datos de tipo NUMBER y DATE de nuevo en un tipo de dato CHARACTER
- A esto se le conoce como conversión de datos implícita





Conversión de Tipo

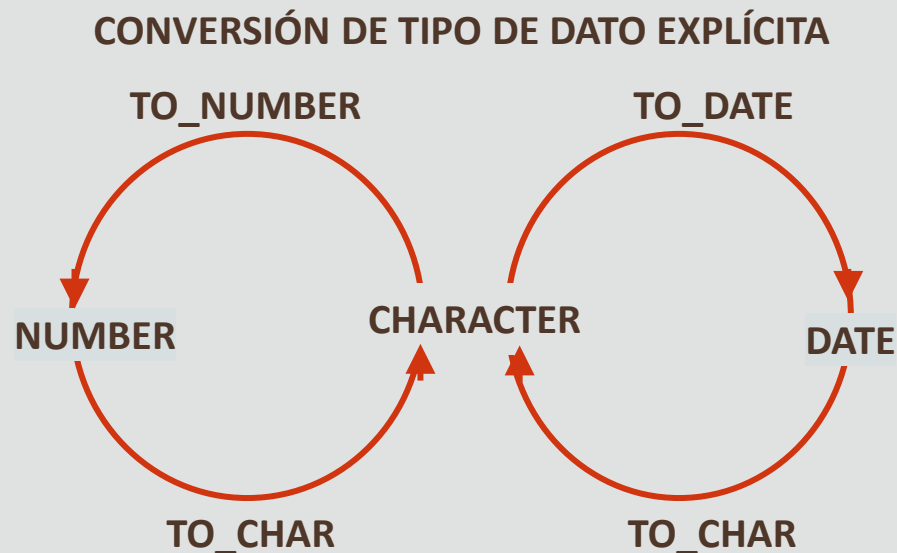
- Aunque esto es una característica útil, siempre es mejor realizar conversiones de tipo de dato explícitas para garantizar la fiabilidad en las sentencias SQL

Conversiones de tipos de dato implícitas

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

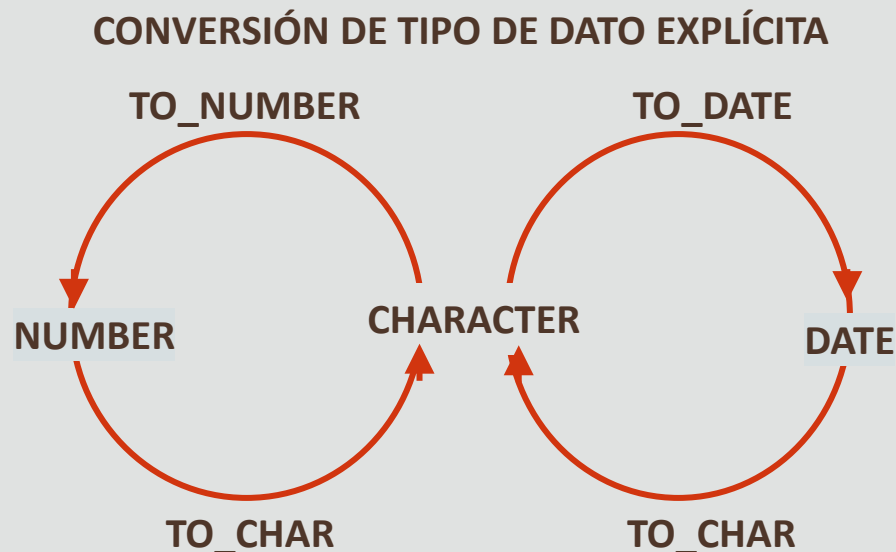
Conversión de Tipo

- Las cuatro funciones de conversión de tipos de dato que aprenderá son:
 - Convertir un tipo de dato de fecha en tipo de dato de carácter
 - Convertir un tipo de dato numérico en tipo de dato de carácter



Conversión de Tipo

- Las cuatro funciones de conversión de tipos de dato que aprenderá son:
 - Convertir un tipo de dato de caracteres en tipo de dato numérico
 - Convertir un tipo de dato de caracteres en tipos de dato de fecha



Conversión de Datos de Fecha en Datos de Caracteres

- Por lo general, es aconsejable que un formato de fecha se convierta de su formato DD-Mes-YYYY por defecto en otro formato especificado por el usuario
- La función que permite llevar esto a cabo es:

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

- El 'modelo de formato' debe estar entre comillas simples y es sensible a mayúsculas/minúsculas
- Separe el valor de fecha del modelo de formato con una coma

Conversión de Datos de Fecha en Datos de Caracteres

- Se puede incluir cualquier elemento de formato de fecha válido
- Utilice sp para deletrear un número
- Utilice th para que el número aparezca como un ordinal (1º, 2º, 3º y así sucesivamente)
- Utilice un elemento fm para eliminar los espacios en blanco o eliminar los ceros iniciales de la salida

Conversión de Datos de Fecha en Datos de Caracteres

YYYY	Año completo en números
YEAR	Año en letra
MM	Valor de dos dígitos del mes
MONTH	Nombre completo del mes
MON	Abreviatura de tres letras del mes
DY	Abreviatura de tres letras del día de la semana
DAY	Nombre completo del día de la semana
DD	Día numérico del mes
DDspth	FOURTEENTH
Ddspth	Fourteenth
ddspth	fourteenth
DDD o DD o D	Día del año, mes o semana
HH24:MI:SS AM	15:45:32 PM
DD "of" MONTH	12 de octubre

- Las tablas muestran los diferentes modelos de formato que se pueden utilizar
- Al especificar elementos temporales, tenga en cuenta que también se puede aplicar formato a las horas (HH), los minutos (MI), los segundos (SS) y AM o PM



Conversión de Datos de Fecha en Datos de Caracteres

- Ejemplos de salida con diferentes modelos de formato:

Ejemplos:	Salida
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 ...

Conversión de Datos de Fecha en Datos de Caracteres

- Ejemplos de salida con diferentes modelos de formato:

Ejemplos:	Salida
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

Conversión de Datos de Fecha en Datos de Caracteres

- Ejemplos de salida con diferentes modelos de formato para el tiempo:

Ejemplos:	Salida
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

Conversión de Datos Numéricos en Datos de Caracteres (VARCHAR2)

- Los números almacenados en la base de datos no tienen formato
- Esto significa que no tienen ningún signo/símbolo de moneda, comas, decimales, ni ningún otro formato
- Para agregar formato, en primer lugar debe convertir el número en un formato de caracteres

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

- La función SQL que se utiliza para convertir un número en un formato de carácter deseado es:

Conversión de Datos Numéricos en Datos de Caracteres (VARCHAR2)

- En la tabla se muestran algunos de los elementos de formato que se pueden utilizar con las funciones TO_CHAR

```
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

Conversión de Datos Numéricos en Datos de Caracteres (VARCHAR2)

- ¿Puede identificar los modelos de formato utilizados para generar la siguiente salida?
 - 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

Conversión de Datos Numéricos en Datos de Caracteres (VARCHAR2)

- Respuestas:

SQL:	Salida
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, '0,009,999') FROM dual;	0004422

Conversión de Caracteres en Números

- Por lo general, es aconsejable convertir una cadena de caracteres en un número. La función necesaria para realizar esta conversión es:

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

- El modelo de formato es opcional, pero se debería incluir si la cadena de caracteres que se va a convertir contiene cualquier carácter que no sean números
- No puede realizar cálculos de forma fiable con datos de caracteres

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

Number
5320



Conversión de Caracteres en Números

- La columna bonus incluye los datos que contienen 4 caracteres y el modelo de formato especifica 3 caracteres, por lo que se devuelve un error

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



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

LAST_NAME	Bonus
Zlotkey	1500
Abel	1700
Taylor	1250

Conversión de Caracteres en Fechas

- Para convertir una cadena de caracteres en un formato de fecha, utilice:

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

- Esta conversión toma una cadena de caracteres que no sea un valor de fecha como, por ejemplo, "3 noviembre de 2001" y lo convierte en un valor de fecha
- El modelo de formato indica al servidor que la cadena de caracteres "se parece a":

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

– Devolverá 03-Nov-2001

Reglas del Modificador fx

- Cuando realiza una conversión de caracteres en fecha, el modificador fx (formato exacto) especifica la coincidencia exacta entre el argumento de carácter y el modelo de formato de fecha
- En el siguiente ejemplo, tenga en cuenta que en "May10 " no hay ningún espacio entre "May" y "10"
- El modelo de formato fx busca la coincidencia con el argumento de caracteres, ya que tampoco tiene ningún espacio entre "Mon" y "DD"

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

CONVERT
10-May-1989

Reglas del Modificador fx

- Las reglas del modificador fx son:
 - La puntuación y el texto entre comillas en el argumento de caracteres deben coincidir con las partes correspondientes del modelo de formato exactamente (excepto en lo que respecta a mayúsculas/minúsculas)
 - El argumento de carácter no puede tener espacios en blanco adicionales
 - Sin fx, Oracle Server ignora los espacios en blanco adicionales
 - Los datos numéricos del argumento de carácter deben tener el mismo número de dígitos que el elemento correspondiente en el modelo de formato
 - Sin fx, los números del argumento de carácter no pueden omitir los ceros iniciales

Reglas del 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

Formato de Fecha RR y Formato de Fecha YY

- Todos los datos de fecha ahora se almacenan utilizando años de cuatro dígitos (YYYY)
- No obstante, algunas bases de datos de legado pueden seguir utilizando el formato de dos dígitos (YY)
- No hace tanto tiempo que hemos cambiado de siglo, de 1900 a 2000
- Este cambio vino acompañado de una gran confusión en función de si una fecha escrita como 02-Jan-98 se interpretaría como 2 de enero de 1998 o 2 de enero de 2098

Formato de Fecha RR y Formato de Fecha YY

- Si los datos que se van a convertir de datos de caracteres a datos de fecha solo contienen un año de dos dígitos, Oracle tiene una forma de interpretar estas fechas con el siglo correcto
- Por ejemplo: '27-Oct-95'

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

Date
27-Oct-2095

- El año de dos dígitos se interpreta como 2095, que puede que no sea lo previsto

Formato de Fecha RR y Formato de Fecha YY

- Si YY se utiliza en el modelo de formato, se asume que el año pertenece al siglo actual
- Si el año de dos dígitos no está en el siglo actual, utilizamos RR

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

Date
27-Oct-1995

- El año de dos dígitos ahora se interpreta como 1995

Algunas Sencillas Reglas

- Si el formato de fecha se especifica con el formato RR, el valor de retorno tiene dos posibilidades, en función del año actual
- Si el año actual está entre 00-49:
 - Fechas a partir de 0-49: la fecha estará en el siglo actual
 - Fechas a partir de 50-99: la fecha estará en el siglo pasado

		Si el año de dos dígitos especificado es:	
		0-49	50-99
Si dos de los dígitos del año actual son:	0-49	La fecha de devolución está en el siglo actual	La fecha de devolución está en el siglo anterior al actual
	50-99	La fecha de devolución está en el siglo posterior al actual	La fecha de devolución está en el siglo actual

Algunas Sencillas Reglas

- Si el año actual está entre 50-99:
 - Fechas a partir de 0-49: la fecha estará en el siglo siguiente
 - Fechas a partir de 50-99: la fecha estará en el siglo actual

		Si el año de dos dígitos especificado es:	
		0-49	50-99
Si dos de los dígitos del año actual son:	0-49	La fecha de devolución está en el siglo actual	La fecha de devolución está en el siglo anterior al actual
	50-99	La fecha de devolución está en el siglo posterior al actual	La fecha de devolución está en el siglo actual

Algunas Sencillas Reglas

- En la siguiente tabla se proporcionan algunos ejemplos de cómo se interpretan YY y RR y, en función del año actual

Año Actual	Fecha Especificada	Formato RR	Formato YY
1995	27-Oct-95	1995	1995
1995	27-Oct-17	2017	1917
2015	27-Oct-17	2017	2017
2015	27-Oct-95	1995	2095

Algunas Sencillas Reglas

- Cuando consulto mi base de datos de empleados utilizando la siguiente sentencia, devuelve todas las filas de la tabla
- Sé que solo existen unos pocos empleados que se contrataron 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 el modelo de formato en la cláusula WHERE utiliza YY, y el año actual es 2015, la consulta devuelve filas con un valor hire_date inferior a 2090

Terminología

- Entre los términos clave utilizados en esta lección se incluyen:
 - CHAR
 - DATE
 - Formato de fecha DD
 - Funciones de conversión
 - fm
 - NUMBER

Terminología

- Entre los términos clave utilizados en esta lección se incluyen:
 - Formato de fecha RR
 - TO_CHAR
 - TO_DATE
 - TO_NUMBER
 - VARCHAR2
 - Modificador fx

Resumen

- En esta lección, debe haber aprendido lo siguiente:
 - Proporcionar un ejemplo de conversión de tipo de dato implícita y explícita y una conversión de tipo de dato implícita
 - Explicar el motivo de la importancia, desde una perspectiva de negocio, de que un lenguaje tenga capacidades de conversión de datos incorporadas
 - Crear una consulta SQL que aplique correctamente las funciones de una sola fila TO_CHAR, TO_NUMBER y TO_DATE para obtener el resultado deseado

Resumen

- En esta lección, debe haber aprendido lo siguiente:
 - Aplicar el modelo de formato de fecha y/o caracteres adecuado para producir una salida deseada
 - Explicar y aplicar el uso de YY y RR para devolver el año correcto como se almacenó en la base de datos



ORACLE

Academy

