TEMA 5: FUNCIONES DE ORACLE

0 INTRODUCCIÓN.

Las funciones de Oracle se pueden utilizar en cualquier sentencia SQL en la que se pueda introducir por sintaxis una referencia a una función (*ref_funcion*), así como en las instrucciones de programas PL/SQL. Se distinguen 5 tipos básicos de funciones en Oracle:

- FUNCIONES ARITMÉTICAS.
- FUNCIONES DE CADENA
- FUNCIONES DE FECHA.
- FUNCIONES DE CONVERSIÓN
- OTRAS FUNCIONES.

0.1 LA TABLA DUAL.

Una función en Oracle sólo puede utilizarse en los contextos comentados anteriormente: dentro de una sentencia SQL o en un programa PL/SQL.

Por ejemplo:

`SELECT ABS(salario)` `FROM EMPLE;`	Dentro de una sentencia SQL (donde la sintaxis lo permita)

No podemos, por ejemplo, probar como funciona una función de forma aislada en un editor de sentencias SQL. Para poder probar el funcionamiento de las funciones, Oracle suministra una tabla especial denominada DUAL. Esta tabla es una tabla de una sola columna (DUMMY), que se puede utilizar para realizar sentencias SQL de prueba:

```
SQL> DESC DUAL;

 Nombre                    ¿Nulo?   Tipo
 ------------------------- -------  ------------
 DUMMY                              VARCHAR2(1)
```

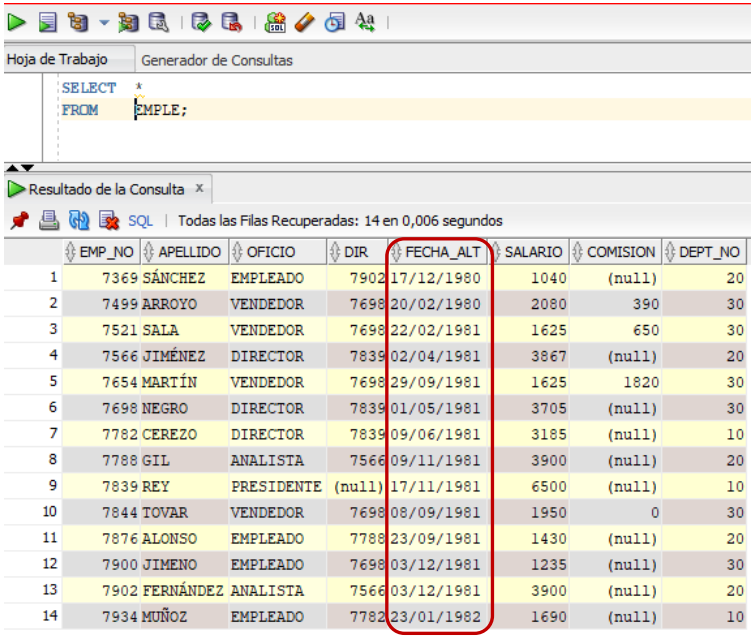
Por ejemplo, si introducimos en un editor de sentencias SQL de Oracle:

`ABS(-5)`	Nos da un error, no puede hacerse así
`SELECT ABS(-5)` `FROM DUAL;`	Ha de utilizarse la función dentro de una sentencia SQL (por ejemplo una SELECT) La tabla DUAL nos permite realizar la prueba dentro de una sentencia SQL

0.2 PARÁMETROS DE SESIÓN.

Antes de explicar este punto es muy importante que entendáis bien la diferencia entre como se almacena internamente un dato (de cualquier tipo) y cuál es su FORMATO de visualización. es decir, como se muestra en la herramienta con la que trabajamos (y por consiguiente como lo interpretamos).

Por ejemplo, Oracle almacena las fechas internamente de una forma determinada (que no viene al caso, para nosotros una ristra de bits), sin embargo, nosotros al visualizarlas lo hacemos con un formato establecido (por ejemplo, dd/mm/yyyy), que en realidad nos hace conceptualizar o entender esa fecha:



Hoja de Trabajo | Generador de Consultas

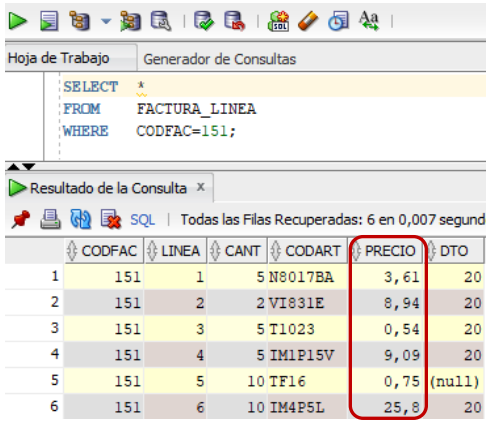
```
SELECT  *
FROM    EMPLE;
```

Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 14 en 0,006 segundos

	EMP_NO	APELLIDO	OFICIO	DIR	FECHA_ALT	SALARIO	COMISION	DEPT_NO
1	7369	SÁNCHEZ	EMPLEADO	7902	17/12/1980	1040	(null)	20
2	7499	ARROYO	VENDEDOR	7698	20/02/1980	2080	390	30
3	7521	SALA	VENDEDOR	7698	22/02/1981	1625	650	30
4	7566	JIMÉNEZ	DIRECTOR	7839	02/04/1981	3867	(null)	20
5	7654	MARTÍN	VENDEDOR	7698	29/09/1981	1625	1820	30
6	7698	NEGRO	DIRECTOR	7839	01/05/1981	3705	(null)	30
7	7782	CEREZO	DIRECTOR	7839	09/06/1981	3185	(null)	10
8	7788	GIL	ANALISTA	7566	09/11/1981	3900	(null)	20
9	7839	REY	PRESIDENTE	(null)	17/11/1981	6500	(null)	10
10	7844	TOVAR	VENDEDOR	7698	08/09/1981	1950	0	30
11	7876	ALONSO	EMPLEADO	7788	23/09/1981	1430	(null)	20
12	7900	JIMENO	EMPLEADO	7698	03/12/1981	1235	(null)	30
13	7902	FERNÁNDEZ	ANALISTA	7566	03/12/1981	3900	(null)	20
14	7934	MUÑOZ	EMPLEADO	7782	23/01/1982	1690	(null)	10

Lo mismo ocurre con los números (su formato de visualización es importante para su interpretación):



Hoja de Trabajo | Generador de Consultas

```
SELECT  *
FROM    FACTURA_LINEA
WHERE   CODFAC=151;
```

Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 6 en 0,007 segund

	CODFAC	LINEA	CANT	CODART	PRECIO	DTO
1	151	1	5	N8017BA	3,61	20
2	151	2	2	VI831E	8,94	20
3	151	3	5	T1023	0,54	20
4	151	4	5	IM1P15V	9,09	20
5	151	5	10	TF16	0,75	(null)
6	151	6	10	IM4P5L	25,8	20

Lo que se debe tener muy claro es que, en ningún caso, los datos se almacenan internamente como se visualizan (formato). En Oracle el formato se configura a través de los llamados **PARÁMETROS DE CONFIGURACIÓN DE SESIÓN**

Por defecto, el formato para: la fecha, la coma decimal, el separador de miles y el símbolo de la moneda, viene definido por el parámetro de sesión **NLS_TERRITORY** que especifica el territorio en el que nos encontramos. Este parámetro se inicializa según la configuración del sistema operativo al arrancar Oracle. Para el idioma español, el valor de este parámetro es: NLS_TERRITORY=SPAIN.

En realidad, el valor de este parámetro determina automáticamente el valor de otros parámetros que especifican independientemente los formatos fijados:

PARÁMETRO	DESCRIPCIÓN
NLS_DATE_FORMAT	Formato de fecha
NLS_DATE_LANGUAGE	Lenguaje utilizado para nombrar los meses y los días de la semana
NLS_NUMERIC_CHARACTERS	Define los caracteres decimal (‘D’) y separador de miles (‘G’).
NLS_ISO_CURRENCY	Especifica el símbolo del territorio. Para España el símbolo es 'ESP'.
NLS_CURRENCY	Especifica el símbolo de la moneda local. Para España es '€'.

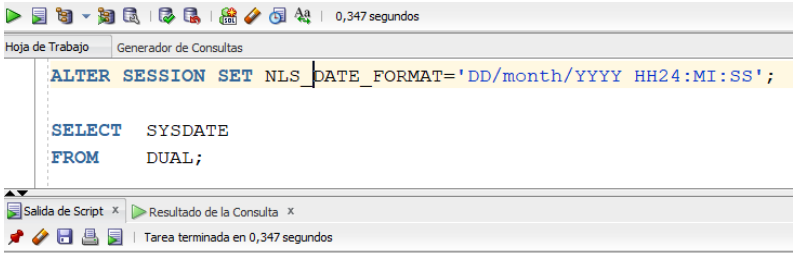
El valor los parámetros de sesión puede ser cambiado utilizando el comando ALTER SESSION. Las configuraciones realizadas mediante este comando tienen efecto hasta que se desconecta de la BD (finaliza la sesión). Al volver a conectar se establecerán de nuevo los formatos por defecto obtenidos de la configuración del sistema operativo.

```
ALTER SESSION SET parameter_name = parameter_value
```

Se deben consultar las ayudas para averiguar la sintaxis correcta de los posibles valores de configuración de un parámetro de sesión (parameter_value). Por ejemplo:

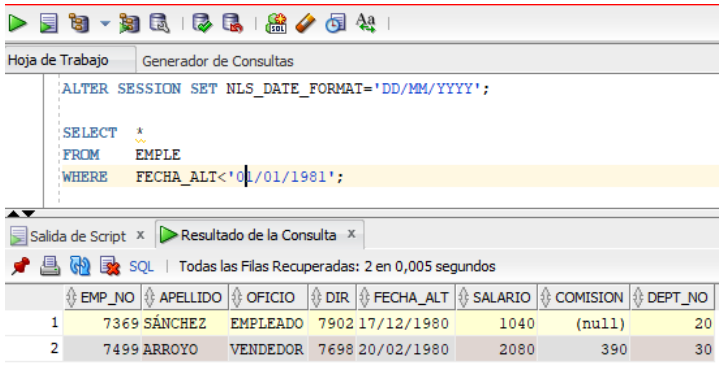
NLS_DATE_FORMAT = "format"/ NLS_DATE_FORMAT = 'format'	En format se deben especificar los caracteres de formato de fecha adecuados.
NLS_NUMERIC_CHARACTERS="DG"/ NLS_NUMERIC_CHARACTERS='DG'	D= carácter separador decimal G= carácter separador de miles
NLS_CURRENCY = currency_symbol	
NLS_ISO_CURRENCY = territory	

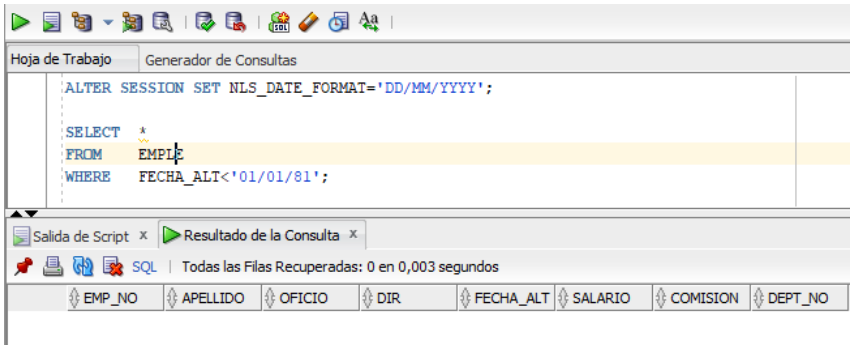
Por ejemplo:



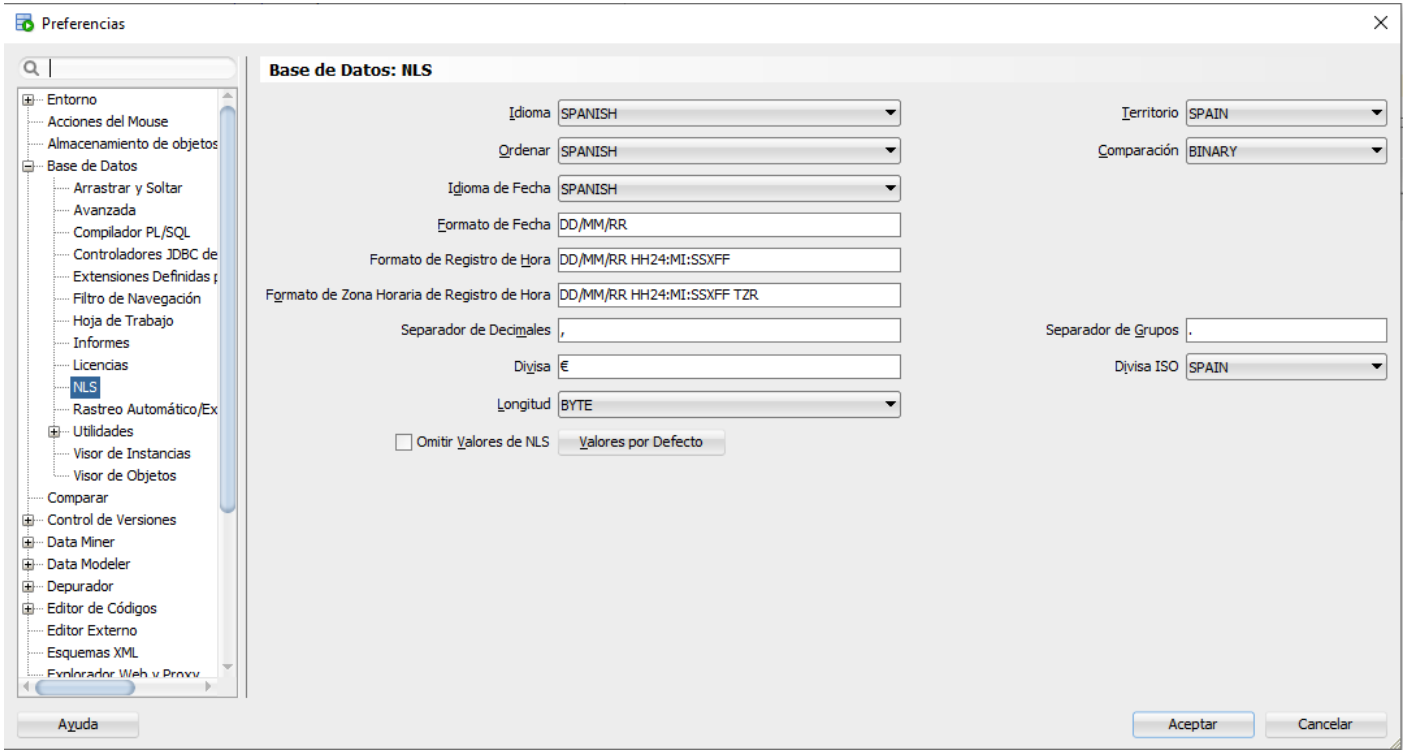

```
Session alterado.


SYSDATE
---------------------------
20/febrero   /2022 08:27:59
```

El valor de los parámetros de configuración de sesión determina la forma en la que Oracle espera los literales de los tipos de datos afectados por el parámetro. Esto más importante de lo que pensamos, sobre todo con las fechas:





En SQL Developer es posible cambiar los valores por defecto de los parámetros de configuración de sesión y dejarlos fijados para todas las sesiones (os recomiendo dejarlos como están para no encontraros con sorpresas al hacer las consultas):



1 FUNCIONES ARITMÉTICAS.

FUNCIONES NO AGREGADAS O ESCALARES: En una SELECT devuelven un valor por cada fila calculado a partir del valor que toma(n) el (los) parámetro(s) en la fila.	
PATRÓN	**ACCIÓN / RESULTADO**
ABS (n)	Devuelve el valor absoluto de n
CEIL (n)	Obtiene el valor entero inmediatamente superior o igual a n
FLOOR (n)	Obtiene el valor entero inmediatamente inferior o igual a n
MOD (m, n)	Devuelve el resto resultante de dividir m entre n (m, n pueden ser reales)
NVL (n, m)	Sustituye el valor nulo por otro valor: si n toma valor nulo lo sustituye por el valor de m. ➢ Puede utilizarse con cualquier tipo de datos: n y m deben ser del mismo tipo
POWER (n, m)	Calcula n^m
TRUNC (n, [,m])	Trunca n con m dígitos decimales: ➢ m<0: trunca m dígitos a la izquierda del punto decimal. ➢ si se omite m es como poner un 0.
ROUND (n [,m])	Redondea n con m dígitos decimales: ➢ m<0: redondea m dígitos a la izquierda del punto decimal. ➢ si se omite m es como poner un 0.
SIGN (n)	Indica el signo de n ➢ n positivo: devuelve 1. ➢ n negativo: devuelve -1.
FUNCIONES AGREGADAS: En una SELECT devuelven un ÚNICO VALOR calculado a partir de los valores de una ref_columna en distintas filas (mayoría estándar de SQL). **NO LAS VEMOS EN ESTE TEMA (TEMA 7. DML 1ª PARTE)**	
COUNT (*)| {SUM| AVG| MAX| MIN| COUNT} ([ALL|DISTINCT] *ref_columna*)	Funciones agregadas estándar de SQL.

2 FUNCIONES DE CADENA (ESCALARES)

FUNCIONES QUE RETORNAN CADENAS:	
CHR (n)	Devuelve el carácter cuyo código ASCII es n
CONCAT (cad1, cad2)	Devuelve la cadena resultado de concatenar cad1 y cad2 ➢ El operador || hace lo mismo
LOWER (cad)	Devuelve la cadena resultado de convertir cad a minúsculas
UPPER (cad)	Devuelve la cadena resultado de convertir cad a mayúsculas
INITCAP (cad)	Devuelve la cadena resultado de formatear cad a tipo titulo: la primera letra de cada palabra a mayúsculas y el resto a minúsculas

LPAD (cad1, n [, cad2])	Devuelve la cadena resultado de rellenar por la IZQUIERDA cad1 con cad2 hasta que cad1 alcance la longitud n: ➢ Si cad2 se omite se asume el carácter blanco
RPAD (cad1, n [, cad2])	Devuelve la cadena resultado de rellenar por la DERECHA cad1 con cad2 hasta que cad1 alcance la longitud n: ➢ Si cad2 se omite se asume el carácter blanco
LTRIM (cad1 [, cad2))	Devuelve la cadena resultado de suprimir por la IZQUIERDA de cad1 los caracteres individuales que se encuentran en cad2: ➢ Se suprime por la izquierda hasta encontrar en cad1 un carácter que no se encuentre en cad2. ➢ **cad2 no se utiliza como cadena completa, sino que se interpreta como conjunto de caracteres.** ➢ Si cad2 se omite se asume el carácter blanco.
RTRIM (cad1 [, cad2))	Devuelve la cadena resultado de suprimir por la DERECHA de cad1 los caracteres individuales que se encuentran en cad2: ➢ Se suprime por la derecha hasta encontrar en cad1 un carácter que no se encuentre en cad2. ➢ **cad2 no se interpreta como cadena completa, sino que se interpreta como conjunto de caracteres.** ➢ Si cad2 se omite se asume el carácter blanco.
REPLACE (cad, cad_busqueda [, cad_sustitucion])	Devuelve la cadena resultado de sustituir en cad cualquier ocurrencia de cad_busqueda por cad_sustitucion ➢ Si cad_sustitucion se omite se asume que se desea borrar cad_busqueda de cad (OJO: borrar NO es sustituir por espacios).
SUBSTR (cad, pos_inicio [, longitud])	Devuelve la cadena resultado de extraer en cad la subcadena desde la posición pos_inicio hasta alcanzar longitud. ➢ Si pos_inicio<0 se asume que se desea comenzar desde el final de cad (PERO longitud SE CUENTA HACIA DELANTE). ➢ Si longitud se omite se devuelve la subcadena completa comenzando desde pos_inicio. ➢ Si longitud<0 error
TRANSLATE (cad1, cad2, cad3)	Devuelve la cadena resultado de sustituir en cad1 los caracteres encontrados en cad2 **por sus correspondientes por posición** en cad3. ➢ Si en la posición de cad3 correspondiente a la de cad2 no hay carácter se asume que se borra de cad1 el carácter que aparece en cad2. ➢ Cualquier carácter que no esté en cad2 permanece como estaba.
FUNCIONES QUE DEVUELVEN VALORES NUMÉRICOS:	
ASCII (cad)	Devuelve el código ASCII del primer carácter de la cadena cad
INSTR (cad1, cad2 [,pos_inicio [,num_ocurrencia]])	Devuelve la posición de la ocurrencia num_ocurrencia de cad2 en cad1, comenzando la búsqueda desde la posición pos_inicio ➢ Si pos_inicio<0 se asume que se desea comenzar desde el final de cad1 (pero devuelve la posición contando desde el principio). ➢ Si se omiten pos_inicio y/o num_ocurrencia se asume que son 1. ➢ Si no se encuentra cad2 en cad1 INSTR devuelve 0
LENGTH (cad)	Devuelve el número de caracteres de cad

3 FUNCIONES PARA MANEJO DE FECHAS (ESCALARES)

SYSDATE	Devuelve la fecha del sistema
ADD_MONTHS (fecha, n)	Devuelve la fecha resultado de incrementar en n meses fecha ➢ n puede ser negativo: en este caso fecha se decrementa n meses.
LAST_DAY (fecha)	Devuelve la fecha correspondiente al último día del mes en el que se ubica fecha
MONTHS_BETWEEN (fecha1, fecha2)	Devuelve la diferencia en meses entre fecha1 y fecha2
NEXT_DAY (fecha, caddia)	Devuelve la fecha del primer día de la semana indicado por caddia después de la fecha indicada por fecha ➢ El día de la semana en caddia se indica con su nombre (según `NLS_DATE_LANGUAGE`) : 'Lunes', 'Martes'...

3.1 USO DE OPERADORES ARITMETICOS CON FECHAS.

OPERACIÓN	DESCRIPCIÓN	TIPO RESULTADO
fecha+n	Añade n días a fecha	FECHA
fecha-n	Resta n días a fecha	FECHA
fecha1-fecha2	Devuelve el nº de días entre fecha1 y fecha2	NUMERICO
fecha+n/24	Añade n horas a fecha	FECHA
fecha-n/24	Resta n horas a fecha	FECHA

3.2 USO DE LAS FUNCIONES TRUNC Y ROUND CON FECHAS.

TRUNC (fecha, ['formato'])	Trunca fecha según la máscara de formato especificada en 'formato' ➢ si se omite 'formato' es como poner dd.
ROUND (fecha, ['formato'])	Redondea fecha según la máscara de formato especificada en 'formato' ➢ si se omite 'formato' es como poner dd.

4 FUNCIONES DE CONVERSIÓN (ESCALARES)

TO_CHAR (dato [, 'formato'])	Transforma un tipo DATE o NUMBER en una cadena de caracteres (tipo VARCHAR2) según el formato especificado ➢ dato puede ser una fecha o un numero. ➢ 'formato' depende del tipo del dato a transformar (ver tablas con máscaras de conversión) ➢ Pueden especificarse en formato cadenas literales entre comillas dobles
TO_NUMBER (cad [, 'formato'])	Convierte la cadena cad a un número (tipo NUMBER) según el formato especificado ➢ cad sólo puede contener números, el carácter decimal y el signo menos a la izquierda.
TO_DATE (cad, [, 'formato'])	Convierte una cadena (tipo CHAR o VARCHAR2) a un valor de tipo DATE según el formato especificado ➢ Si se omite 'formato' la especificación de la cadena cad debe coincidir con el formato de fecha del sistema

Los caracteres necesarios para especificar las máscaras de formato en las funciones de conversión deben consultarse siempre en las ayudas de cada versión. No obstante, al finalizar este documento se ofrece un resumen orientativo.

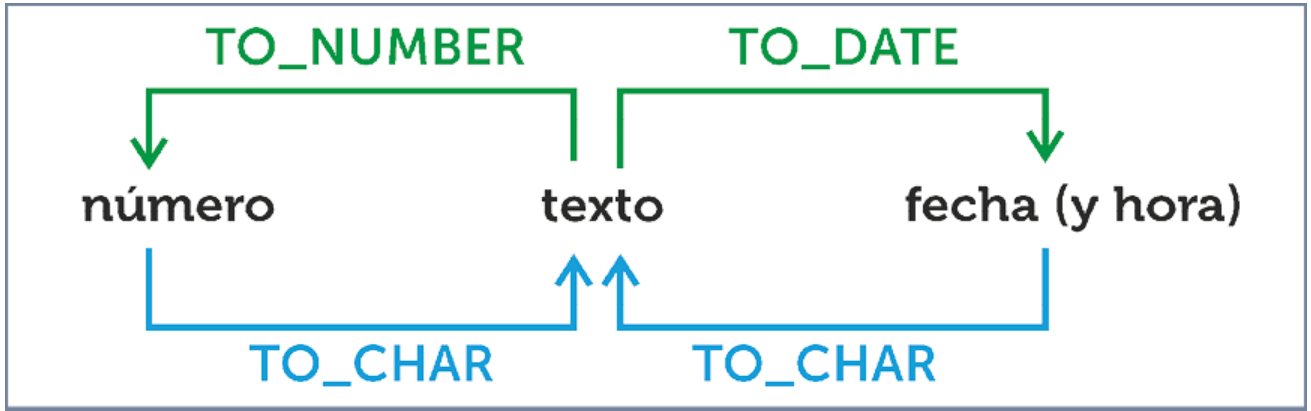


Ilustración 4. Uso de las funciones de conversión

5 OTRAS FUNCIONES

USER	Devuelve el nombre del usuario actual (con el que estamos conectados en la sesión)
DECODE (ref_columna, valor1, cod1, valor2, cod2..., v_defecto)	Sustituye o codifica unos valores por otros: si ref_columna es igual a cualquier valor de la lista: valor1, valor2, ..., etc. se sustituye por su correspondiente código de la lista cod1, cod2, ..., etc. ➢ En caso que ref_columna no coincida con ningún valor de la lista se sustituye por el valor señalado como v_defecto.
VSIZE (ref_columna)	Devuelve el tamaño en bytes de ref_columna ➢ Si es ref_columna es nulo devuelve nulo
DUMP (cad [,formato [, pos_inicio, [longitud]]])	Codifica la subcadena extraída de cad desde pos_inicio tomando longitud caracteres según el formato especificado: ➢ 8: octal ➢ 10: decimal ➢ 16: hexadecimal ➢ 17: ASCII o EBCDIC

5.1 FUNCIONES PARA UTILIZAR EXPRESIONES REGULARES. CARÁCTER INFORMATIVO (NO LAS VAMOS A UTILIZAR)

REGEXP_LIKE (cad, patron [, correspondencia])	Parecido al operador LIKE, pero realiza una correspondencia de expresiones regulares en lugar de una correspondencia de patrones sencillos
REGEXP_REPLACE (cad_busqueda, patron [,cad_sustitucion [, pos_inicio [, ocurrencia [, correspondencia]]]])	Busca un patrón de expresión regular y lo sustituye por una cadena de sustitución
REGEXP_INSTR (cad, patron [, pos_inicio [, ocurrencia [, return_option [, correspondencia]]]])	Busca en una cadena un patrón de expresión regular y devuelve la posición en la que se encuentra la correspondencia
REGEXP_SUBSTR (cad, patron [, pos_inicio [, ocurrencia [, correspondencia]]])	Busca un patrón de expresión regular dentro de una cadena especificada y devuelve la subcadena con la correspondencia

Las expresiones regulares son muy potentes y todo un mundo. **NO LAS VAMOS A UTILIZAR**, pero quiero que sepáis que estas funciones existen por si las necesitáis en el futuro.

6 TABLAS ORIENTATIVAS CARACTERES DE FORMATO.

6.1 CARACTERES DE FORMATO DE FECHA.

Máscaras de formato numéricas	
cc o scc	Valor del siglo
y, yyy ó sy,yyy	Año con coma, con o sin signo
yyyy	Año sin signo
yyy	Últimos tres dígitos del año
yy	Últimos dos dígitos del año
y	Último dígito del año
q	Número del trimestre
ww	Número de semana del año
w	Número de semana del mes
mm	Número de mes
ddd	Número de día del año
dd	Número de día del mes
d	Número de día de la semana
hh ó hh12	Hora (1-12)
hh24	Hora (1-24)
mi	Minutos
ss	Segundos
sssss	Segundos transcurridos desde medianoche
j	Juliano

Máscaras de formato de caracteres	
syear ó year	Año en inglés (ej: *nineteen-eighty-two*)
month	Nombre del mes (ENERO)
mon	Abreviatura de tres letras del nombre del mes (ENE)
day	Nombre del día de la semana (LUNES)
dy	Abreviatura de tres letras del nombre del día (LUN)
a.m. o p.m.	Muestra a.m. ó p.m. dependiendo del momento del día
b.c. o a.d.	Indicador para el año (antes de Cristo o después de Cristo)

6.1.1 USO DE LOS FORMATOS RR/ RRRR PARA LOS AÑOS.

Existe la posibilidad de utilizar los formatos RR/ RRRR en lugar de YY/YYYY para especificar años numéricos. Los formatos RR/RRRR surgieron con el objetivo de flexibilizar el siglo cuando no se escriben los 2 primeros números del año. **Aprovecho para remarcar que es conveniente escribir siempre los 4 dígitos del año en los literales de tipo fecha para evitar sorpresas no deseadas**. Este apartado es la demostración más evidente.

FORMATO RR VS YY

En principio estos formatos son similares, y permiten indicar el año en una fecha utilizando únicamente 2 dígitos. La diferencia entre ambos radica en cómo interpreta Oracle este año, **concretamente los dos primeros dígitos del año**:

FORMATO	2 PRIMEROS DÍGITOS (INTERPRETADOS POR ORACLE)	EJEMPLO
YY	2 PRIMEROS DÍGITOS AÑO ACTUAL	**DD/MM/YY** 21/06/23= 21/06/**20**23 21/06/99= 21/06/**20**99
RR	• RR= 1..49→ 2 PRIMEROS DÍGITOS AÑO ACTUAL • RR= 50..99 →2 PRIMEROS DÍGITOS AÑO ANTERIOR ACTUAL	**DD/MM/RR** 21/06/23= 21/06/**20**23 21/06/99= 21/06/**19**99

Con ambos formatos es correcto escribir los 4 dígitos completos. En ambos formatos se interpretarán literalmente. Es altamente recomendable.

FORMATO RRRR VS YYYY

Estos formatos se utilizan para indicar el año con 4 dígitos, **por lo que deberían indicarse los literales siempre con 4 dígitos**, cuidado con ellos si utilizamos 2 dígitos:

FORMATO	2 PRIMEROS DÍGITOS (INTERPRETADOS POR ORACLE)	EJEMPLO
YYYY	00 (con este formato espera los literales con 4 dígitos en el año y no se los estamos indicando)	**DD/MM/YY** 27/02/23= 27/02/**00**23 27/02/99= 27/02/**00**99
RRRR	• RR= 1..49→ 2 PRIMEROS DÍGITOS AÑO ACTUAL • RR= 50..99 →2 PRIMEROS DÍGITOS AÑO ANTERIOR ACTUAL	**DD/MM/RR** 27/02/23= 27/02/**20**23 27/02/99= 27/02/**19**99

MI CONSEJO ☺ ¡¡¡ESCRIBID SIEMPRE LOS LITERALES DE TIPO FECHA CON 4 DÍGITOS EN EL AÑO!!!

6.2 CARACTERES DE FORMATO NUMÉRICOS.

9	999	Devuelve el valor con el número especificado de dígitos. Si es positivo, deja un espacio.
		Devuelve el valor con el número especificado de dígitos con el signo menos si es negativo.
		Si el valor tiene ceros a la izquierda, los deja en blanco, excepto si el valor es 0.
		`SQL> SELECT TO_CHAR(1,'999'), TO_CHAR(-1,'999'), TO_CHAR(01,'999') , TO_CHAR(0,'999') FROM DUAL;`
		`TO_C TO_C TO_C TO_C`
		`---- ---- ---- ----`
		`1 -1 1 0`

0	9990
	999
		9990 - Muestra un 0 si el valor es 0.
		0999 - Devuelve el valor dejando ceros al principio.
		`SQL> SELECT TO_CHAR(10,'0999'), TO_CHAR(10,'9990') ,TO_CHAR(10,'990090') FROM DUAL;`
		`TO_CH TO_CH TO_CHAR`
		`----- ----- -------`
		`0010 10 0010`

$	$9999	Devuelve el valor con el signo dólar a la izquierda.
		`SQL> SELECT TO_CHAR(10,'$9999'),TO_CHAR(10,'$009'), TO_CHAR(10,'99$') FROM DUAL;`
		`TO_CHA TO_CH TO_C`
		`------ ----- ----`
		`$10 $010 $10`

B	B999	Muestra un espacio en blanco si el valor es 0. Es el formato por omisión.
		`SQL> SELECT TO_CHAR(0,'B999'), TO_CHAR(5,'B999') FROM DUAL;`
		`TO_C TO_C`
		`---- ----`
		`5`

MI	999MI	Si el número es negativo, el signo menos sigue al número. Por omisión, el signo se pone a la izquierda.
		`SQL> SELECT TO_CHAR(-55,'999MI'), TO_CHAR (55, '999MI') FROM DUAL;`
		`TO_C TO_C`
		`---- ----`
		`55- 55`

S	S999
	999S
		'S' representa el signo. Devuelve el valor con el signo '+' si el valor es positivo o con el signo '_' si es negativo.
		`SQL> SELECT TO_CHAR(-55,'999S'), TO_CHAR(-55, 'S999'), TO_CHAR(55,'S999'), TO_CHAR(55,'999S') FROM DUAL;`
		`TO_C TO_C TO_C TO_C`
		`---- ---- ---- ----`
		`55- -55 +55 55+`

PR	9999PR	Los números negativos se muestran entre estos símbolos: < >.
		`SQL> SELECT TO_CHAR(-55,'9999PR'), TO_CHAR(55, '9999PR') FROM DUAL;`
		`TO_CHA TO_CHA`
		`------ ------`
		`<55> 55`

D	99D99	Devuelve el carácter decimal en la posición especificada.
		`SQL> SELECT TO_CHAR(34.55,'99D99') FROM DUAL;`
		`TO_CHA`
		`------`
		`34,55`

G	9G999	Devuelve el carácter de grupo (carácter de los miles) en la posición especificada.
		`SQL> SELECT TO_CHAR(1234,'9G999') FROM DUAL;`
		`TO_CHA`
		`------`
		`1.234`
		`SQL> SELECT TO_CHAR(123456.98, '999G999D99') FROM DUAL;`
		`TO_CHAR(123`
		`---------`
		`123.456,98`

C	C999	Devuelve el símbolo ISO del territorio en la posición especificada.
		`SQL> SELECT TO_CHAR(123,'C999') "ISO" FROM DUAL;`
		`ISO`
		`------`
		`EUR123`

L	L999
	999L
		Devuelve el símbolo de la moneda local en la posición indicada.
		`SQL> SELECT TO_CHAR(123,'L999') "MONEDA" FROM DUAL;`
		`MONEDA`
		`------`
		`€ 123`

, (coma)	9,999	Devuelve la coma en la posición especificada (carácter de los miles).
		`SQL> SELECT TO_CHAR(1234,'9,999') FROM DUAL;`
		`TO_CHA`
		`------`
		`1,234`

. (punto)	99.99	Devuelve el punto decimal en la posición especificada.
		`SQL> SELECT TO_CHAR(12.34,'99.99') FROM DUAL;`
		`TO_CHA`
		`------`
		`12.34`
		`SQL> SELECT TO_CHAR(12345.67,'99,999.99') FROM DUAL;`
		`TO_CHAR(12`
		`---------`
		`12,345.67`

V	999V99	Devuelve el valor multiplicado por 10n, donde 'n' es el número de nueves después 'V'.
		`SQL> SELECT TO_CHAR(123.45,'999V99'), TO_CHAR (123,'999V99') FROM DUAL;`
		`TO_CHA TO_CHA`
		`------ ------`
		`12345 12300`

EEEE	9.9EEEE	Devuelve el valor usando notación científica.
		`SQL> SELECT TO_CHAR(12345,'9.9EEEE') FROM DUAL;`
		`TO_CHAR(1`
		`---------`
		`1.2E+04`

RN
rn
	RN	Devuelve el valor en números romanos. 'RN' devuelve el valor en mayúsculas y 'rn', en minúsculas.
		`SQL> SELECT TO_CHAR(12,'RN') , TO_CHAR(12,'rn') FROM DUAL;`
		`TO_CHAR(12,'RN' TO_CHAR(12,'RN'`
		`---------- ----------`
		`XII xii`

FM	FM90.9	Devuelve el valor alineado a la izquierda.
		`SQL>SELECT TO_CHAR(12.8,'FM90.9'),TO_CHAR(12, 'FM99'), TO_CHAR(-12,'FM99') FROM DUAL;`
		`TO_CH TO_ TO_`
		`----- --- ---`
		`12.8 12 -12`

Element	Example	Description
, (comma)	`9,999`	Returns a comma in the specified position. You can specify multiple commas in a number format model. **Restrictions:** • A comma element cannot begin a number format model. • A comma cannot appear to the right of a decimal character or period in a number format model.
. (period)	`99.99`	Returns a decimal point, which is a period (.) in the specified position. **Restriction:** You can specify only one period in a number format model.
$	`$9999`	Returns value with a leading dollar sign.
0	`0999` `9990`	Returns leading zeros. Returns trailing zeros.
9	`9999`	Returns value with the specified number of digits with a leading space if positive or with a leading minus if negative. Leading zeros are blank, except for a zero value, which returns a zero for the integer part of the fixed-point number.
B	`B9999`	Returns blanks for the integer part of a fixed-point number when the integer part is zero (regardless of zeros in the format model).
C	`C999`	Returns in the specified position the ISO currency symbol (the current value of the `NLS_ISO_CURRENCY` parameter).
D	`99D99`	Returns in the specified position the decimal character, which is the current value of the `NLS_NUMERIC_CHARACTER` parameter. The default is a period (.). **Restriction:** You can specify only one decimal character in a number format model.
EEEE	`9.9EEEE`	Returns a value using in scientific notation.
G	`9G999`	Returns in the specified position the group separator (the current value of the `NLS_NUMERIC_CHARACTER` parameter). You can specify multiple group separators in a number format model. **Restriction:** A group separator cannot appear to the right of a decimal character or period in a number format model.
L	`L999`	Returns in the specified position the local currency symbol (the current value of the `NLS_CURRENCY` parameter).
MI	`9999MI`	Returns negative value with a trailing minus sign (-). Returns positive value with a trailing blank. **Restriction:** The MI format element can appear only in the last position of a number format model.
PR	`9999PR`	Returns negative value in <angle brackets>. Returns positive value with a leading and trailing blank. **Restriction:** The PR format element can appear only in the last position of a number format model.
RN rn	`RN` `rn`	Returns a value as Roman numerals in uppercase. Returns a value as Roman numerals in lowercase. Value can be an integer between 1 and 3999.

S	`S9999` `9999S`	Returns negative value with a leading minus sign (-). Returns positive value with a leading plus sign (+). Returns negative value with a trailing minus sign (-). Returns positive value with a trailing plus sign (+). **Restriction:** The S format element can appear only in the first or last position of a number format model.
TM	`TM`	The text minimum number format model returns (in decimal output) the smallest number of characters possible. This element is case insensitive. The default is TM9, which returns the number in fixed notation unless the output exceeds 64 characters. If the output exceeds 64 characters, then Oracle Database automatically returns the number in scientific notation. **Restrictions:** • You cannot precede this element with any other element. • You can follow this element only with one 9 or one E (or e), but not with any combination of these. The following statement returns an error: • `SELECT TO_CHAR(1234, 'TM9e') FROM DUAL;`
U	`U9999`	Returns in the specified position the Euro (or other) dual currency symbol (the current value of the `NLS_DUAL_CURRENCY` parameter).
V	`999V99`	Returns a value multiplied by 10^n (and if necessary, round it up), where n is the number of 9's after the `V`.
X	`XXXX` `xxxx`	Returns the hexadecimal value of the specified number of digits. If the specified number is not an integer, then Oracle Database rounds it to an integer. **Restrictions:** • This element accepts only positive values or 0. Negative values return an error. • You can precede this element only with 0 (which returns leading zeroes) or FM. Any other elements return an error. If you specify neither 0 nor FM with X, then the return always has 1 leading blank.

Number Format Elements