**TO\_NUMBER( string1 [, format\_mask] [, nls\_language] )**

This parameter is used to add extra information to the function in order for the value to be displayed. It works similar to the TO\_CHAR function but has a few differences.

The expression added here will represent the thousand group marker, decimal marker, and the currency symbols to be used. The expression for this parameter can contain one or more of the following parameters, separated by commas:

* NLS\_CURRENCY *symbol*
* NLS\_ISO\_CURRENCY *territory*
* NLS\_NUMERIC\_CHARACTERS *dg*

With these statements:

* *symbol* represents the local currency symbol and must be less than or equal to 10 characters
* *territory* is a text expression that identifies the territory whose ISO currency symbol is used
* *dg* represents two characters, which are single-byte characters, used for the decimal marker and thousands group marker.

**Format Mask**

|  |  |
| --- | --- |
| 9 | Single number character. Removes leading and trailing zeroes. Includes a leading hyphen (-) for negative numbers. |
| 0 | Single number character. Includes leading and trailing zeroes. |
| FM | Indicates that the format should suppress all leading and trailing blanks |
| $ | Include a leading $ sign |
| B | Changes the ‘0’ format and replace leading 0’s with blank spaces. |
| S | Include both positive (+) and negative (-) signs |
| PR | Indicate sign of the number with angle brackets (<>) |
| MI | Indicate the minus sign |
| RN or rn | Displays the number in Roman numerals (upper or lower case) |
| D | Indicates the position of the decimal point (.) |
| G | Adds a group separator (often a comma) in a position |
| , | Adds a comma in a position |
| . | Adds a period in a position |
| C | Adds the ISO currency symbol |
| L | Adds the local currency symbol |
| EEEE | Displays the value using scientific notation |
| TM | Displays the Text Minimum in TM9(the default) or TME (Scientific notation) |
| U | Displays in the specified position the Euro (or other) dual currency symbol. |
| V | Displays the value multiplied by 10n, where n is the number of 9’s after the letter V. The value may be rounded up if necessary |
| X | Returns the hexadecimal value of the specified number of digits. |

SELECT

TO\_NUMBER('5467.12')

FROM DUAL;

SELECT

TO\_NUMBER('5467.12', '999999.99')

FROM DUAL;

SELECT

TO\_NUMBER('4687841', '9999999')

FROM DUAL;

SELECT TO\_NUMBER('$65.169', 'L99.999')

FROM DUAL;

SELECT

TO\_NUMBER('123,456,789', '999,999,999')

FROM DUAL;

SELECT

TO\_NUMBER('$17 218,00', 'L999G999D00',' NLS\_NUMERIC\_CHARACTERS='', ''')

FROM DUAL;

TO\_DATE (date\_text [, format\_mask] [, nls\_date\_language])

The date\_text is the date you want to convert, which is in some kind of text or character format. You can optionally specify the format mask (which is the format that this date value was provided in), and the nls\_date\_langauge is used for dates in different languages or countries.

SELECT TO\_DATE('21-JAN-2017', 'DD-MON-YYYY') AS converted\_date

FROM dual;

SELECT TO\_DATE('20170115\_142309', 'YYYYMMDD\_HH24MISS') AS converted\_date

FROM dual;

NEW\_TIME (input\_date, input\_timezone, output\_timezone)

SELECT SYSDATE,

NEW\_TIME(SYSDATE, 'GMT', 'PST') AS converted\_time

FROM dual;

FROM\_TZ

This function converts a TIMESTAMP value and a specified TIME ZONE to a TIMESTAMP WITH TIME ZONE value.

If you need a value that’s in a TIMESTAMP WITH TIME ZONE format, then this is the function to use.

The syntax is:

FROM\_TZ (timestamp\_value, timezone\_value)

SELECT FROM\_TZ(TIMESTAMP '2017-04-19 07:13:50', '-9:00') AS from\_tz\_output

FROM dual;

## SYS\_EXTRACT\_UTC

The SYS\_EXTRACT\_UTC will extract or convert the specified date and time into a UTC (also known as GMT) date and time.

You specify a data type that has a timezone, and a TIMESTAMP is returned that shows the time in UTC time.

SELECT SYS\_EXTRACT\_UTC(TIMESTAMP '2017-05-15 19:10:45 +10:00') AS utc\_time

FROM dual;

This converts the specified time, which is +10 UTC, into a UTC time. The UTC time is 10 hours before the specified time.

## SESSIONTIMEZONE

The SESSIONTIMEZONE function returns the timezone offset of your session, in the format of [+|-]TZH:TZM, or a time zone region name.

This can be helpful to know, especially if you’re doing a lot of work with dates and work on databases in different time zones.

SELECT SESSIONTIMEZONE

FROM dual;

### **How Can You Change the Session Time Zone?**

ALTER SESSION SET TIME\_ZONE = '+8:0';

### **What’s The Difference Between SESSIONTIMEZONE and CURRENT\_TIMESTAMP?**

Both of these functions look at the time of the session. But there are some differences.

* CURRENT\_TIMESTAMP returns the entire date and time, including the time zone.
* SESSIONTIMEZONE returns only the time zone.

Sometimes, SESSIONTIMEZONE may be easier to use than trying to perform string manipulation on the CURRENT\_TIMESTAMP function.

## DBTIMEZONE

The DBTIMEZONE function returns the timezone offset of the database, in the format of [+|-]TZH:TZM, or a time zone region name.

It’s helpful when working with dates to know what timezone the database is in. This is easier than using one of the other date functions, or performing an extraction from a TIMESTAMP WITH TIME ZONE value.

This DBTIMEZONE function is how to check the timezone in Oracle database 11g – and all other database versions.

SELECT DBTIMEZONE

FROM dual;

### **How Can You Change the Database Time Zone?**

The database time zone is actually the time zone of the operating system of the server it runs on.

So, you can’t specifically change the time zone on the database.

SELECT SESSIONTIMEZONE FROM DUAL;

### **What’s the Difference Between Oracle DBTIMEZONE and SESSIONTIMEZONE in Oracle?**

The difference between these two functions is:

* DBTIMEZONE returns the time zone of the database server.
* SESSIONTIMEZONE returns the time zone of your session.

These two values may be different, if you’re in a different location to your database.

### **What’s The Difference Between Oracle DBTIMEZONE vs SYSTIMESTAMP?**

The difference between DBTIMEZONE and SYSTIMESTAMP is:

* DBTIMEZONE returns just the timezone offset of the database (e.g. +7:00).
* SYSTIMESTAMP returns the timestamp with time zone of the database (e.g. 13-JUL-2016 08:49:02 AM +7:00)

So, the SYSTIMESTAMP includes more information than the DBTIMEZONE function.

**Date Format**

**To\_char**

|  |  |
| --- | --- |
| **Parameter** | **Explanation** |
| YEAR | Year, spelled out in full words |
| YYYY | 4-digit year |
| YYY | Last 3 digits of year |
| YY | Last 2 digits of year |
| Y | Last digit of year |
| IYY | Last 3 digits of ISO year |
| IY | Last 2 digits of ISO year |
| I | Last digit of ISO year |
| IYYY | 4-digit year, which is based on the ISO standard |
| RRRR | This format accepts a 2-digit year, and returns a 4-digit year. If the provided value is between 0 and 49, it will return a year greater than or equal to 2000. If the provided value is between 50 and 99, it will return a year less than 2000 |
| Parameter | Explanation |
| Q | Quarter of year, from 1 to 4. JAN to MAR = 1 |
| MM | Month, from 01 to 12. JAN = 01 |
| MON | Abbreviated name of month. |
| MONTH | Name of month, padded with blanks to length of 9 characters. |
| RM | Roman numeral month, from I to XII. JAN = I. |

**Week**

|  |  |
| --- | --- |
| **Parameter** | **Explanation** |
| WW | Week of year, from 1 to 53. Week 1 starts on the first day of the year, and continues to the seventh day of the year. |
| W | Week of month, from 1 to 5. Week 1 starts on the first day of the month and ends on the seventh. |
| IW | Week of year, from 1 to 52 or 1 to 53, based on the ISO standard. |

|  |  |
| --- | --- |
| **Parameter** | **Explanation** |
| D | Day of week, from 1 to 7. |
| DAY | Name of day. |
| DD | Day of month, from 1 to 31. |
| DDD | Day of year, from 1 to 366. |
| DY | Abbreviated name of day. |
| J | Julian day, which is the number of days since January 1, 4712 BC. |

**Time**

|  |  |
| --- | --- |
| **Parameter** | **Explanation** |
| HH | Hour of day, from 1 to 12. |
| HH12 | Hour of day, from 1 to 12. |
| HH24 | Hour of day, from 0 to 23. |
| MI | Minute, from 0 to 59 |
| SS | Second, from 0 to 59 |
| SSSSS | Seconds past midnight, from 0 to 86399. |
| FF | Fractional seconds. This uses a value from 1 to 9 after FF, to indicate the number of digits in the fractional seconds (e.g. FF7) |

|  |  |
| --- | --- |
| **Parameter** | **Explanation** |
| AM, A.M., PM, or P.M. | Meridian indicator |
| AD or A.D | AD indicator |
| BC or B.C. | BC indicator |
| TZD | Daylight savings information |
| TZH | Time zone hour. |
| TZM | Time zone minute. |
| TZR | Time zone region. |