

Table of Contents

1. Cast.....	1
2. Formatting numbers with To_Char.....	1
3. Date formats	2
4. Formatting date values with To_Char	3

Oracle will do a lot of implicit data conversions for you, but there may be times when you need to do an explicit data type conversion. Some of the intrinsic functions are very "fussy" about the data type of their arguments; sorting is different if it is done alphabetically or numerically. Union queries also may need conversion functions to produce union compatible columns. The conversion functions are: CAST, TO_CHAR, and TO_DATE (To_Date is discussed with the temporal functions)

1. Cast

CAST is a general purpose conversion function. It can be used to convert between the built-in data types.

These cast functions work.

```
Cast ( '345' as Number)      → 345
Cast ( '345.678' as Number (8,2) )    → 345.68
Cast ( '345678' as Number (8,2) )    → 345678
```

The following will fail. The first specifies too narrow a target; the second has a string argument that Oracle cannot convert.

```
Cast ( '345678' as Number (5,2) ) →
ORA-01438: value larger than specified precision allowed for this column

cast('VII' as number)
ORA-01722: invalid number
```

2. Formatting numbers with To_Char to get a string

To_Char is used to convert a numeric or date expression into a string; you can supply a format to be used for the conversion. This can be used if you want to have consistently formatted output. The result is enclosed within two uprights so that you can see where the spaces are in the result.

Examples of Using ' ' TO_CHAR (NumExp, 'FormatString') ' '	Result
TO_CHAR (3.5, '9999')	4
TO_CHAR (3.5, '0999')	0004
TO_CHAR (3.5, '999.99')	3.50
TO_CHAR (3.5, '\$999.99')	\$3.50
TO_CHAR (3.5, '\$099.99')	\$003.50
TO_CHAR (34567.5, '099.99')	#####
TO_CHAR (34567.5, '999,999.99')	34,567.50
TO_CHAR (34567.5, '9,999,99.99')	345,67.50
TO_CHAR (345, '999')	345

TO_CHAR (345, 'FM999')	345
TO_CHAR (345, 'S999')	+345
TO_CHAR (345, '999S')	345+
TO_CHAR (345, '999V99')	34500
TO_CHAR (-345, '999V99')	-34500
TO_CHAR (345, 'RN')	CCCXLV
TO_CHAR (345, 'rn')	cccxlv

Numeric Formats to use with TO_CHAR		Used to
Element	Example	
9	999.99	Display the specified number of digits; does not display a leading zero; there will be a leading space or minus sign
0	0999	Displays a leading zero if applicable
\$	\$999,999.00	Floating dollar symbol; moves to the left
L	L9999.00	The Local currency symbol
.	9.999	Decimal point; only one decimal point allowed
,	99,999,999	Comma ; you need to position this carefully; you can have multiple commas but they must appear to the left of any decimal point
PR at the end of the format	999.00PR	To display negative numbers inside angle brackets <99.00> is the negative number -99.00
EEEE	9.99EEEE	For scientific notation
FM	FM99.9	Suppresses the leading space
S	S999 999S	Displays a leading (or trailing) plus or minus sign.
MI	999MI	Display a minus sign or trailing blank in the last position.
V	999V99	Returns the number multiplied by 10 ⁿ , where n is the number of digits after the V symbol.

3. Date formats

You need to keep in mind that the format used to display a date is a string; it is a representation of a stored date value. Suppose we have a datetime value of July 4, 2012 midnight. We could display this value as "July 4, 2012", "2012-07-04", or , "2012/7/4" and these would each be a different string representation of the same value .

There is a default format for displaying and entering date values. The default format is commonly of the form '19-AUG-02' represented as a DD-MON-RR format. This means that dates entered with a two digit year have the first two digits of the year calculated based on the last two digits of the current year and the last two digits of

the specified year. It is safer to specify a 4-digit year format and enter dates with a 4-digit year. Since several tools can change default settings, you should not rely on a default date format.

Since we enter and display dates in a variety of formats, there are conversion functions to handle the differences.

Oracle DATE values always include a data and a time component. Suppose you create a table d_sysdate with a single date attribute and insert the value of sysdate.

```
create table z_sysdate (coll date);
insert into z_sysdate values (sysdate);
```

Wait a minute and then run the following query.

```
select * from z_sysdate
where coll = sysdate;
```

No rows will be returned since the time portions will not match.

If your date values are not time sensitive, then you could use date functions (such as TRUNC) to handle this issue when testing a data value against another date value.

4. Formatting date values with To_Char to get a string

The TO_CHAR function is used to specify a display format for a date value. This is a partial list of formats; consult your text or the manuals for a more complete list.

Examples run on Thursday June 03, 2004.

Examples	Result
TO_CHAR (SYSDATE, 'YYYY-MM-DD')	2004-06-03
TO_CHAR (SYSDATE, 'MON DD, YYYY')	JUN 03, 2004
TO_CHAR (SYSDATE, 'MM')	06
TO_CHAR (SYSDATE, 'YYYY')	2004
TO_CHAR (SYSDATE, 'D')	5 {Thurs is the 5 th day of the week}
TO_CHAR (SYSDATE, 'Month DD, yyyy') The month name takes 9 spaces	June 03, 2004
"fm" is used to turn on and off leading zeros and blank padding	
TO_CHAR (SYSDATE, 'fmMonth DD, yyyy') Now the month name gets only the space it needs	June 3, 2004
TO_CHAR (SYSDATE, 'fmMonth fmDD, yyyy') The month name is not padded, but the day number is.	June 03, 2004
TO_CHAR (SYSDATE, 'MON')	JUN
TO_CHAR (SYSDATE, 'DDD')	155
TO_CHAR (SYSDATE, 'DY')	THU
TO_CHAR (SYSDATE, 'day')	thursday
TO_CHAR (SYSDATE, ' DAY ')	THURSDAY
TO_CHAR (SYSDATE, 'Mon Ddth')	Jun 03rd
TO_CHAR (SYSDATE, 'Mon Ddsp')	Jun Three

TO_CHAR (SYSDATE, 'Mon DDspth')	Jun THIRD
TO_CHAR (SYSDATE, 'Mon Ddth Yyyysp')	Jun 03rd Two Thousand Four
TO_CHAR (SYSDATE, 'J')	2453160 {Julian day- start Dec 31 4712BC}

Time formats

If we display the value of sysdate, the default display format show only the day month and year. But sysdate does have a time component.

```
select sysdate from dual;
```

```
SYSDATE
-----
19-FEB-14
```

If we want to see the time, we need to include time formats.

```
select to_char(sysdate, 'HH24 MI SS') as "Time" from dual;
```

```
Time
-----
19  40  49
```

Wait a few moments and run the query again.

```
Time
-----
19  41  28
```

You can combine the date and time formats.

```
select to_char(sysdate, 'YYYY-Mon-DD HH24:MI') as "Date and Time" from dual;
```

```
Date and Time
-----
2014-Feb-19 19:43
```

```
select to_char(sysdate, 'YYYY-Mon-DD HH24 "Hours and" MI "Minutes"')
as "Date and Time"
from dual;
```

```
Date and Time
-----
2014-Feb-19 19 Hours and 45 Minutes
```