

# [三思笔记]日期时间及数字的格式化参数大全

**2007.12.13**

经常遇到有朋友询问关于日期时间格式化的问题(或可以通过格式化轻易解决), 经过参考 Oracle SQL Reference 官方文档, 并附上详细的应用示例综合成文, 希望能够对大家学习和使用有所帮助。本篇可视为"oracle 著名及非著名函数介绍"的补充!

环境:

```
SQL> select *from v$version;
```

BANNER

```
-----  
Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 - Prod  
PL/SQL Release 10.2.0.3.0 - Production  
CORE      10.2.0.3.0      Production  
TNS for 32-bit Windows: Version 10.2.0.3.0 - Production  
NLSRTL Version 10.2.0.3.0 - Production
```

```
SQL> SELECT SESSIONTIMEZONE FROM DUAL;
```

SESSIONTIMEZONE

```
-----  
+08:00
```

```
SQL> show parameter nls;
```

NAME	TYPE	VALUE
-----	-----	-----
nls_date_format	string	yyyy-mm-dd hh24:mi:ss
nls_language	string	SIMPLIFIED CHINESE

## 一、日期及时间

格式串      说明及示例

-----  
-/,,:      指定返回字符串分隔符

例如:

```
SQL> select to_char(sysdate,'yyyy/mm/dd') from dual;
```

```
TO_CHAR(SYSDATE,'YYYY/MM/DD')
```

```
2007/12/14
SQL> select to_char(sysdate,'yyyy-mm-dd') from dual;

TO_CHAR(SYSDATE,'YYYY-MM-DD')
-----
2007-12-14
```

AD/A.D. 公元标识

```
例如:
SQL> select to_char(sysdate,'AD YYYY-MM-DD') from dual;

TO_CHAR(SYSDATE,'ADYYYY-MM-DD')
-----
公元 2007-12-14
```

BC/B.C. 公元标识

```
例如
SQL> select to_char(sysdate,'BC YYYY-MM-DD') from dual;

TO_CHAR(SYSDATE,'BCYYYY-MM-DD')
-----
公元 2007-12-14
```

AM/A.M. 子午线标识

```
例如:
SQL> select to_char(sysdate,'YYYY-MM-DD AM hh24:mi:ss') from dual;

TO_CHAR(SYSDATE,'YYYY-MM-DDAMH')
-----
2007-12-14 下午 13:41:58
```

PM/P.M. 子午线标识

```
例如:
SQL> select to_char(sysdate,'YYYY-MM-DD PM hh24:mi:ss') from dual;

TO_CHAR(SYSDATE,'YYYY-MM-DDPMH')
-----
2007-12-14 下午 13:42:23
```

CC/SCC 世纪标识, S 前缀指定如遇公元前的显示, 会在显示前加(-)

提示:

- 如果年份中最后两位数字在 01 到 99(含)之间, 则返回值等于年份前两位+1
- 如果年份中最后两位数字是 00, 则返回值与年份前两位相同

例如:

```
SQL> select to_char(to_date('2000-8-3','YYYY-MM-DD'),'cc') from dual;
```

```
TO_CHAR(TO_DATE('2000-8-3','YY
```

```
-----
```

```
20
```

```
SQL> select to_char(to_date('2001-8-3','YYYY-MM-DD'),'cc') from dual;
```

```
TO_CHAR(TO_DATE('2001-8-3','YY
```

```
-----
```

```
21
```

D 指定日期在周中的数值(范围: 1-7)

例如:

```
SQL> select to_char(to_date('2007-12-14','yyyy-mm-dd'),'D') from dual;
```

```
TO_CHAR(TO_DATE('2007-12-14','
```

```
-----
```

```
6
```

DD 指定日期在当月的天数(范围: 1-31)

例如:

```
SQL> select to_char(to_date('2007-12-14','yyyy-mm-dd'),'DD') from dual;
```

```
TO_CHAR(TO_DATE('2007-12-14','
```

```
-----
```

```
14
```

DDD 指定日期在当年中的天数(范围: 1-366)

例如:

```
SQL> select to_char(to_date('2007-12-14','yyyy-mm-dd'),'DDD') from dual;
```

```
TO_CHAR(TO_DATE('2007-12-14','
```

```
-----
```

```
348
```

DAY 指定日期在周中的名称

例如:

```
SQL> select to_char(to_date('2007-12-14','yyyy-mm-dd'),'Day') from dual;
```

```
TO_CHAR(TO_DATE('2007-12-14','
```

```
-----
```

```
星期五
```

**MON** 返回指定日期的月份简写

例如:

```
SQL> select to_char(sysdate,'MON') from dual;
```

```
TO_CHAR(SYSDATE,'MON')
```

```
-----
```

```
12 月
```

与 DY 同, 中文看不出简写效果。

**MONTH** 返回指定日期的月份全称

例如:

```
SQL> select to_char(sysdate,'MONTH') from dual;
```

```
TO_CHAR(SYSDATE,'MONTH')
```

```
-----
```

```
12 月
```

**YEAR/SYEAR** 返回字符型年, 此处 s 前缀与 SCC 处相同

例如:

```
SQL> select to_char(sysdate,'year') from dual;
```

```
TO_CHAR(SYSDATE,'YEAR')
```

```
-----
```

```
two thousand seven
```

**YYYY/SYYYY** 返回数字型年, 此处 s 前缀与 SCC 处相同

例如:

```
SQL> select to_char(sysdate,'yyyy') from dual;
```

```
TO_CHAR(SYSDATE,'YYYY')
```

```
-----
```

```
2007
```

**MM** 返回指定日期的月份(范围: 1-12)

例如:

```
SQL> select to_char(sysdate,'MM') from dual;
```

```
TO_CHAR(SYSDATE,'MM')
```

```
-----
```

```
12
```

**HH** 小时(范围: 1-12)

例如:

```
SQL> select to_char(sysdate,'am hh') from dual;
```

```
TO_CHAR(SYSDATE,'AMHH')
```

```
-----
```

```
下午 02
```

HH12 小时(范围: 1-12)

例如:

```
SQL> select to_char(sysdate,'am hh12') from dual;
```

```
TO_CHAR(SYSDATE,'AMHH12')
```

```
-----
```

```
下午 02
```

HH24 小时(范围: 0-23)

例如:

```
SQL> select to_char(sysdate,'am hh24') from dual;
```

```
TO_CHAR(SYSDATE,'AMHH24')
```

```
-----
```

```
下午 14
```

MI 返回指定时间的分钟(范围 0-59)

例如:

```
SQL> select to_char(sysdate,'MI') from dual;
```

```
TO_CHAR(SYSDATE,'MI')
```

```
-----
```

```
05
```

SS 返回指定时间的秒数(范围: 0-59)

例如:

```
SQL> select to_char(sysdate,'ss') from dual;
```

```
TO_CHAR(SYSDATE,'SS')
```

```
-----
```

```
34
```

FF[1-9] 返回毫秒数, 可指定长度 1-9, 默认 6 位

例如:

```
SQL> select to_char(systimestamp,'ff4') from dual;
```

```
TO_CHAR(S
```

```
-----
```

```
6710
```

Y,YYY 返回有逗号分隔显示的年

例如:

```
SQL> select to_char(sysdate,'y,yyy') from dual;
```

```
TO_CHAR(SYSDATE,'Y,YYY')
```

```
-----
```

```
2,007
```

I/Y/IYY/IYYY 返回 ISO 标准年

例如:

```
SQL> select to_char(to_date('187-4-3','yyyy-mm-dd'),'IYYY') from dual;
```

```
TO_CHAR(TO_DATE('187-4-3','YYY
```

```
-----
```

```
0187
```

Y/YY/YYYY 以指定长度返回日期的年份

例如:

```
SQL> select to_char(sysdate,'yy') from dual;
```

```
TO_CHAR(SYSDATE,'YY')
```

```
-----
```

```
07
```

W 返回指定日期在当月的第 X 周(范围: 1-5)

例如:

```
SQL> select to_char(to_date('2007-6-18','yyyy-mm-dd'),'w') from dual;
```

```
TO_CHAR(TO_DATE('2007-6-18','Y
```

```
-----
```

```
3
```

WW 返回指定日期在当年中的第 X 周(范围: 1-53)

例如:

```
SQL> select to_char(to_date('2007-6-18','yyyy-mm-dd'),'ww') from dual;
```

```
TO_CHAR(TO_DATE('2007-6-18','Y
```

```
-----
```

```
25
```

IW 指定日期在当年中第 X 周(范围: 1-52 或 1-53(闰年))

例如:

```
SQL> select to_char(sysdate,'IW') from dual;
```

```
TO_CHAR(SYSDATE,'IW')
```

```
-----
```

```
50
```

RM 以罗马数字形式返回月份(范围 I-XII)

例如:

```
SQL> select to_char(to_date('2007-11-3','yyyy-mm-dd'),'RM') from dual;
```

```
TO_CHAR(TO_DATE('2007-11-3','Y
```

```
-----
```

```
XI
```

Q 返回指定日期的季度(范围: 1-4)

例如:

```
SQL> select to_char(to_date('2007-2-3','yyyy-mm-dd'),'q') from dual;
```

```
TO_CHAR(TO_DATE('2007-2-3','YY
```

```
-----
```

```
1
```

J 自公元前 4712 年 1 月 1 日到指定日期的总天数

例如:

```
SQL> select to_char(sysdate,'J') from dual;
```

```
TO_CHAR(SYSDATE,'J')
```

```
-----
```

```
2454449
```

SSSSS 返回自午夜到指定时间共逝去的秒数(范围: 0-86399)

例如:

```
SQL> select to_char(sysdate,'sssss') from dual;
```

```
TO_CHAR(SYSDATE,'SSSSS')
```

```
-----
```

```
55141
```

DL 根据数据库 `nls_date_format` 参数中的设置及 `nls_language`, `nls_territory` 参数中的格式返回长日期形式, 有点儿绕, 呵呵你试试就知道了

例如:

```
SQL> select to_char(sysdate,'DL') from dual;
```

```
TO_CHAR(SYSDATE,'DL')
```

```
-----
```

2007 年 12 月 14 日 星期五

DS 与上相同返回短日期形式

例如:

```
SQL> select to_char(sysdate,'Ds') from dual;
```

```
TO_CHAR(SYSDATE,'DS')
```

```
-----
```

```
2007-12-14
```

TS 与 DL, DS 参数类似, 不过 ts 返回的是时间

例如:

```
SQL> select to_char(sysdate,'ts') from dual;
```

```
TO_CHAR(SYSDATE,'TS')
```

```
-----
```

```
下午 3:21:20
```

DY 返回日期简写

例如:

```
SQL> alter session set nls_language='SIMPLIFIED CHINESE';
```

会话已更改。

```
SQL> select to_char(sysdate,'dy') from dual;
```

```
TO_CHAR(SYSD
```

```
-----
```

```
星期五
```

嗯? 怎么没变? 黑黑中文没法简了嘛, 这样试试~~

```
SQL> alter session set nls_language='AMERICAN';
```

Session altered.

```
SQL> select to_char(sysdate,'dy') from dual;
```

```
TO_CHA
```

```
-----
```

```
fri
```

RR RR 参数与 YY 参数很类似, 但取值规则略有不同:

如果输入的两位数是 00-49:

如果当前年份属于 00-49, 则返回成本世纪所在年份;

如果当前年份属于 50-99, 则返回成下世纪所在年份。



如果输入的两位数是 50-99:

如果当前年份属于 00-49, 则返回成上世纪所在年份

如果当前年份属于 50-99, 则返回成本世界所在年份

注: 这种自动转换只适用于 String -> Date 的转换, 如果是 Date -> String, 那就按照 Date 存储的值来转换, 此时再用 RR/RRRR 已经没有意义了, 因为 Oracle 内部存储的年份是四位数的。

例如:

```
SQL> select sysdate from dual;
```

```
SYSDATE
```

```
-----
```

```
2007-12-18 15:11:39
```

```
SQL> select to_date('12','rr') from dual;
```

```
TO_DATE('12','RR')
```

```
-----
```

```
2012-12-01 00:00:00
```

```
SQL> select to_date('51','rr') from dual;
```

```
TO_DATE('51','RR')
```

```
-----
```

```
1951-12-01 00:00:00
```

RRRR 输入值可以是 2 位也可以是 4 位, 2 位时规则与 RR 完全相同。

例如:

```
SQL> select to_date('97-11-3','rrrr-mm-dd') from dual;
```

```
TO_DATE('97-11-3','RRRR-MM-DD')
```

```
-----
```

```
1997-11-3
```

```
SQL> select to_date('17-11-3','rrrr-mm-dd') from dual;
```

```
TO_DATE('17-11-3','RRRR-MM-DD')
```

```
-----
```

```
2017-11-3
```

提示: FM 及 FX 用于 to\_\* 函数中控制空白填充或精确检查, 其提供的是一个格式化方案, 并且具备相当的持久能力, 比如当你第一次调用之后即开始生效, 后续再调用时即使不指定 FM/FX, 也仍然会生效。当你再次调用时失效, 第三次调用时又开始生效, 以此类推(有点儿绕, 不明白的看下面的示例, 示例还看不明白的直接问我, 问我也问不明白地话.....黑, 估计你也用不着, 算鸟)。。。。。

FM 填充模式。

Oracle 默认使用空白字符填充被格式化的元素, 确保宽度一致。例如:

例如:

```
SQL> select to_char(to_date('2007-3-1','yyyy-mm-dd'),'month') mm,length(to_char(to_date('2007-3-1','yyyy-mm-dd'),'month')) ll from dual;
```

MM	LL
march	9

-----

```
SQL> select to_char(to_date('2007-12-1','yyyy-mm-dd'),'month') mm,length(to_char(to_date('2007-12-1','yyyy-mm-dd'),'month')) ll from dual;
```

MM	LL
december	9

-----

如上例可以看到, 不管几月份, 其返回所占长度是一致的。

- 如果是字符格式化, 指定 **fm** 参数后将仅返回指定属性实际所占长度(不再以空格填充)。

例如:

```
SQL> select to_char(to_date('2007-3-1','yyyy-mm-dd'),'fmmonth') mm,length(to_char(to_date('2007-3-1','yyyy-mm-dd'),'fmmonth')) ll from dual;
```

MM	LL
march	5

-----

```
SQL> select to_char(to_date('2007-12-1','yyyy-mm-dd'),'fmmonth') mm,length(to_char(to_date('2007-12-1','yyyy-mm-dd'),'fmmonth')) ll from dual;
```

MM	LL
december	8

-----

与上对比, 看出区别来了吧。

- 如果是数值格式化, **fm** 参数会自动舍弃被格式化元素的前置 0。

例如:

```
SQL> alter session set nls_date_format='fmyyyy-mm-dd hh24:mi:ss';
```

Session altered.

```
SQL> select to_date('2007-8-1 03:03:04','yyyy-mm-dd hh24:mi:ss') from dual;
```

TO\_DATE('2007-8-103

```
-----  
2007-8-1 3:3:4
```

```
SQL> alter session set nls_date_format='yyyy-mm-dd hh24:mi:ss';
```

```
Session altered.
```

```
SQL> select to_date('2007-8-1 03:03:04','yyyy-mm-dd hh24:mi:ss') from dual;
```

```
TO_DATE('2007-8-103
```

```
-----  
2007-08-01 03:03:04
```

```
SQL> alter session set nls_date_format='fmyyyy-mm-fmdd hh24:fmmi:ss';
```

```
Session altered.
```

```
SQL> select to_date('2007-8-1 03:03:04','yyyy-mm-dd hh24:mi:ss') from dual;
```

```
TO_DATE('2007-8-103
```

```
-----  
2007-8-01 03:3:4
```

FX

精确匹配。使用 FX 参数之后：

- 字符及对应格式必须严格一一对应，甚至连分隔符也需要相符。
- 不允许有多余空格。
- 数值参与格式需要完全对应(或通过 fm 参数自动补足 0)

例如：

```
SQL> select to_date('2007-9-26','yyyy-mm-dd') from dual;
```

```
TO_DATE('2007-9-26'
```

```
-----  
2007-09-26 00:00:00
```

```
SQL> select to_date('2007-9-26','fxyyyy-mm-dd') from dual;
```

```
select to_date('2007-9-26','fxyyyy-mm-dd') from dual
```

```
*
```

```
ERROR at line 1:
```

```
ORA-01862: the numeric value does not match the length of the format item
```

```
SQL> select to_date('2007-09-26','fxyyyy-mm-dd') from dual;
```

```
TO_DATE('2007-09-26
```

```
-----  
2007-09-26 00:00:00
```

```
SQL> select to_date('2007-9-26','fxyyyy-fmmm-dd') from dual;
```

```
TO_DATE('2007-9-26'
```

```
-----  
2007-09-26 00:00:00
```

```
SQL> select to_date('2007-9-26','fxyyyy/fmmm/dd') from dual;
```

```
select to_date('2007-9-26','fxyyyy/fmmm/dd') from dual
```

```
*
```

```
ERROR at line 1:
```

```
ORA-01861: literal does not match format string
```

X

本地的默认分隔符

例如:

```
SQL> select to_char(systimestamp,'hh24:mi:ssxff') from dual;
```

```
TO_CHAR(SYSTIMESTAMP,'HH24:MI:
```

```
-----  
16:21:12.609000
```

以下四项为时区相关参数，明确表示某一个时区，应该联合使用 TZH TZM 或者 TZR TZD  
TZD/TZR

例如:

```
SQL> select to_char(systimestamp,'tztz') from dual;
```

```
TO_CHAR(SYSTIMESTAMP,'TZR:TZD')
```

```
-----  
+08:00:
```

TZH/TZM

例如:

```
SQL> select to_char(systimestamp,'tzh:tzm') from dual;
```

```
TO_CHA
```

```
-----  
+08:00
```

应用示例:

1、与指定日期在同一周，周 1 的日期是:

```
SQL> select to_date('2007-12-6','yyyy-mm-dd')
```

```
2 - to_char(to_date('2007-12-6', 'yyyy-mm-dd'), 'd') + 2
3 from dual;
```

```
TO_DATE('2
```

```
-----
```

```
2007-12-03
```

```
SQL> select trunc(to_date('2007-12-6', 'yyyy-mm-dd'), 'd') + 1 rn from dual;
```

```
TRUNC(TO_DATE('2007-12-6','YYY
```

```
-----
```

```
2007-12-3
```

```
SQL> select next_day(to_date('2007-12-6', 'yyyy-mm-dd') - 7, '星期一') from dual;
```

```
NEXT_DAY(TO_DATE('2007-12-6','
```

```
-----
```

```
2007-12-3
```

## 2、指定日期当月最后一个星期天：

```
SQL> select max(rn)
```

```
2 from (select to_date('2007-12-' || rownum, 'yyyy-mm-dd') rn
```

```
3 from dual
```

```
4 connect by rownum <=
```

```
5 (to_char(to_date('2007-12-31', 'yyyy-mm-dd'), 'dd')) a
```

```
6 where to_char(a.rn, 'd') = 1
```

```
7 ;
```

```
MAX(RN)
```

```
-----
```

```
2007-12-30
```

```
SQL> select next_day(trunc(to_date('2007-12-31', 'yyyy-mm-dd'), 'month')+31-7,'星期日') from dual;
```

```
NEXT_DAY(TRUNC(TO_DATE('2007-1
```

```
-----
```

```
2007-12-30
```

## 3、今年第 127 天是几号：

```
SQL> select to_date('127','ddd') from dual;
```

```
TO_DATE('1
```

```
-----
```

```
2007-05-07
```

## 二、数值

### 格式串

说明及示例

-----  
在指定位置附加逗号(便于阅读), 注意逗号不能出现于格式字串首位, 并且对于小数点的数字无效

例如:

```
SQL> select to_char(389999.00,'99,999,999') from dual;
```

```
TO_CHAR(389
```

```
-----
```

```
389,999
```

. 按指定格式显示小数点及小数点后位置(指定数字没有小数的话, 以 0 补足)

例如:

```
SQL> select to_char(18,'999.000') from dual;
```

```
TO_CHAR(
```

```
-----
```

```
18.000
```

\$ 字符前附加\$符

例如:

```
SQL> select to_char(18,'$999') from dual;
```

```
TO_CH
```

```
-----
```

```
$18
```

```
SQL> select to_char(18,'999$') from dual;
```

```
TO_CH
```

```
-----
```

```
$18
```

0 字符头部或尾部附加 0

例如:

```
SQL> select to_char(18.0,'0999') from dual;
```

```
TO_CH
```

```
-----
```

```
0018
```

9 与上类似以指定长度返回数值格式, 不过如果被格式化的数值长度短于指定格式, 则以空格补足。

例如:

```
SQL> select to_char(18.0,'9999') from dual;
```

```
TO_CH
```

```
-----
```

```
18
```

B 当整数部分是 0 时返回空格(即使存在 0 格式串也忽略)。

例如:

```
SQL> select to_char(0.18,'B0999') from dual;
```

```
TO_CH
```

```
-----
```

C 返回 ISO 默认指定的标识(默认标识可参见:NLS\_ISO\_CURRENCY)

例如:

```
SQL> select to_char(128.18,'C999999.99') from dual;
```

```
TO_CHAR(128.18,'C
```

```
-----
```

```
CNY128.18
```

D 返回指定的小数点形式(默认符: .)(默认标识可参见:NLS\_NUMERIC\_CHARACTER)

例如:

```
SQL> select to_char(128.18,'9999D99') from dual;
```

```
TO_CHAR(
```

```
-----
```

```
128.18
```

```
SQL> alter session set nls_numeric_characters='.';
```

会话已更改。

```
SQL> select to_char(128.18,'9999D99') from dual;
```

```
TO_CHAR(
```

```
-----
```

```
128:18
```

EEEE 使用科学记数法显示

例如:

```
SQL> select to_char(128.18,'9EEEE') from dual;
```

```
TO_CHAR
```

```
-----
```

```
1E+02
```

G            与逗号功能类似，用指定字符分隔字符串（默认符：,）（默认标识可参见:NLS\_NUMERIC\_CHARACTER）。

例如：

```
SQL> select to_char(11128.18,'99G999') from dual;
```

```
TO_CHAR
```

```
-----
```

```
11,128
```

```
SQL> alter session set nls_numeric_characters='.';
```

会话已更改。

```
SQL> select to_char(11128.18,'99G999') from dual;
```

```
TO_CHAR
```

```
-----
```

```
11"128
```

L            指定位置显示本地默认符号（默认值见：NLS\_CURRENCY）

例如：

```
SQL> select to_char(999,'L999') from dual;
```

```
TO_CHAR(999,'L
```

```
-----
```

```
¥999
```

```
SQL> alter session set nls_currency='haha';
```

会话已更改。

```
SQL> select to_char(999,'L999') from dual;
```

```
TO_CHAR(999,'L
```

```
-----
```

```
haha999
```

MI           负值在尾部显示-号，正值在尾部附加空格。另：该格式符必须处于格式串尾部

例如：



```
SQL> select to_char(-18,'999MI') from dual;
```

```
TO_C
```

```
----
```

```
18-
```

```
SQL> select to_char(18,'999MI') from dual;
```

```
TO_C
```

```
----
```

```
18
```

PR

负值以<>尖括号包括，正值在头尾部各附加空格。另：该格式符必须处于格式串尾部

例如：

```
SQL> select to_char(-18,'999PR') from dual;
```

```
TO_CH
```

```
----
```

```
<18>
```

```
SQL> select to_char(18,'999PR') from dual;
```

```
TO_CH
```

```
----
```

```
18
```

RN/rn

返回罗马数字形式。注：大写格式串返回大写罗马数字，小写格式串返回小写罗马数字；)

例如：

```
SQL> select to_char(18,'RN') from dual;
```

```
TO_CHAR(18,'RN'
```

```
-----
```

```
XVIII
```

```
SQL> select to_char(18,'rn') from dual;
```

```
TO_CHAR(18,'RN'
```

```
-----
```

```
xviii
```

S

显示数值正负值符号。可出现于格式串首部或尾部。

例如：

```
SQL> select to_char(18,'S9999') from dual;
```

```
TO_CH
```

```
-----
```

```
+18
```

```
SQL> select to_char(18,'9999S') from dual;
```

```
TO_CH
```

```
-----
```

```
18+
```

TM 字符格式转换(标准数值与科学计数法)，可以通过附加 9 或 e 控制输出，默认是 tm9，如果输出超过 64 个字符，则数据库自动转换成科学计数法显示。该格式串不能与其它的数值型格式串同时使用。

例如：

```
SQL> select TO_CHAR(128,'tme') from dual;
```

```
TO_CHAR(128,'TME')
```

```
-----
```

```
1.28E+02
```

```
SQL> select TO_CHAR(1.28E+02,'tm') from dual;
```

```
TO_CHAR(1.28E+02,'TM')
```

```
-----
```

```
128
```

U 与 L 类似，在指定位置显示货币单位(默认值见：NLS\_DUAL\_CURRENCY)

例如：

```
SQL> select TO_CHAR(128,'999U') from dual;
```

```
TO_CHAR(128,'9
```

```
-----
```

```
128¥
```

```
SQL> alter session set nls_dual_currency='aaa';
```

会话已更改。

```
SQL> select TO_CHAR(128,'999U') from dual;
```

```
TO_CHAR(128,'9
```

```
-----
```

```
128aaa
```

V 数值+10 的 n 次方，n=V 后 9 的个数

例如:

```
SQL> select TO_CHAR(128,'999V999') from dual;
```

TO\_CHAR

-----

128000

X

返回指定数值对应的 16 进制数，必须是正数。如果指定数值非整的话，oracle 会自动 round 取整。

例如:

```
SQL> select TO_CHAR(16,'XXXX') from dual;
```

TO\_CH

-----

10

[\[三思笔记\]RMAN 高级应用之 Duplicate 复制数据库](http://www.itpub.net/906598.html)

[\*\*http://www.itpub.net/906598.html\*\*](http://www.itpub.net/906598.html)

[\[三思笔记\]RHEL AS4 下升级 oracle10g 到 10.2.0.3](http://www.itpub.net/896394.html)

[\*\*http://www.itpub.net/896394.html\*\*](http://www.itpub.net/896394.html)

[\[三思笔记\]RHEL AS4 下安装 32 位 oracle10g](http://www.itpub.net/884137.html)

[\*\*http://www.itpub.net/884137.html\*\*](http://www.itpub.net/884137.html)

[\[三思笔记\]Statspack 初步学和用](http://www.itpub.net/857807.html)

[\*\*http://www.itpub.net/857807.html\*\*](http://www.itpub.net/857807.html)

[\[三思笔记\]oracle 著名及非著名函数介绍](http://www.itpub.net/843333.html)

[\*\*http://www.itpub.net/843333.html\*\*](http://www.itpub.net/843333.html)

[\[三思笔记\]一步一步学 rman](http://www.itpub.net/810100.html)

[\*\*http://www.itpub.net/810100.html\*\*](http://www.itpub.net/810100.html)

[\[三思笔记\]学习动态性能表](http://www.itpub.net/782892.html)

[\*\*http://www.itpub.net/782892.html\*\*](http://www.itpub.net/782892.html)