【函数】Oracle 函数系列（2）--数学函数及日期函数

1.1 BLOG 文档结构图

1.2 前言部分

1.2.1 导读和注意事项

各位技术爱好者，看完本文后，你可以掌握如下的技能，也可以学到一些其它你所不知道的知识，~O(∩_∩)O~：

① 数学函数

② trunc 和 round 函数

③ 常用日期函数

Tips:

① 本文在 itpub（http://blog.itpub.net/26736162）、博客园（http://www.cnblogs.com/lhrbest）和微信公众号（xiaomaimiaolhr）上有同步更新。

② 文章中用到的所有代码、相关软件、相关资料及本文的 pdf 版本都请前往小麦苗的云盘下载，小麦苗的云盘地址见：http://blog.itpub.net/26736162/viewspace-1624453/。

③ 若网页文章代码格式有错乱，请下载 pdf 格式的文档来阅读。

④ 在本篇 BLOG 中，代码输出部分一般放在一行一列的表格中。其中，需要特别关注的地方我都用灰色背景和粉红色字体来表示，比如在下边的例子中，thread 1 的最大归档日志号为 33，thread 2 的最大归档日志号为 43 是需要特别关注的地方；而命令一般使用黄色背景和红色字体标注；对代码或代码输出部分的注释一般采用蓝色字体表示。

```
  List of Archived Logs in backup set 11
  Thrd Seq     Low SCN    Low Time            Next SCN   Next Time
  ---- ------- ---------- ------------------- ---------- ---------
  1    32      1621589    2015-05-29 11:09:52 1625242    2015-05-29 11:15:48
  1    33      1625242    2015-05-29 11:15:48 1625293    2015-05-29 11:15:58
  2    42      1613951    2015-05-29 10:41:18 1625245    2015-05-29 11:15:49
  2    43      1625245    2015-05-29 11:15:49 1625253    2015-05-29 11:15:53
[ZHLHRDB1:root]:/>lsvg -o
T_XLHRD_APP1_vg
rootvg
[ZHLHRDB1:root]:/>
00:27:22 SQL> alter tablespace idxtbs read write;
====》2097152*512/1024/1024/1024=1G
```

本文如有错误或不完善的地方请大家多多指正，ITPUB 留言或 QQ 皆可，您的批评指正是我写作的最大动力。

1.2.2 本文简介

最近一段时间比较忙，就给大家分享一些简单的函数吧。**本部分分享完后就给大家分享有关锁的内容，请大家持续关注小麦苗，谢谢。**

之前发布了 Oracle 函数系列（1）--字符函数（http://blog.itpub.net/26736162/viewspace-2126927/），今天给大家分享的是 Oracle 函数系列（2）--数学函数及日期函数。

1.3 数学函数

数学函数的输入参数和返回值的数据类型都是数字类型的。数学函数包括 cos， cosh，exp，ln，log，sin，sinh，sqrt，tan，tanh，acos，asin，atan，round，如下所示：

- round(n,[m])

  该函数用于执行四舍五入，如果省掉 m，则四舍五入到整数， 如果 m 是正数，则四舍五入到小数点的 m 位后。如果 m 是负数，则四舍五入到小数点的 m 位前。
- trunc(n,[m]) ---可以对数值和 date 类型的数据截取，具体见下面的解析。
- floor(n) 返回小于或是等于 n 的最大整数
- ceil(n) 返回大于或是等于 n 的最小整数
- mod(m,n):取余（ANSI 标准中规定取模运算的符号为%在一些解释器中被函数 MOD 所取代）
- sign(m)：根据参数 m 的值是 0，正数还是负数依次返回 0，1，-1
- abs(n)： 返回数字 n 的绝对值

- acos(n)： 返回数字的反余弦值
- asin(n)： 返回数字的反正弦值
- atan(n)： 返回数字的反正切值
- cosh()：双曲余弦
- sinh()：双曲正弦
- tanh()；双曲正切
- sin(n)；正弦
- cos(n)：余弦
- exp(n)： 返回 e 的 n 次幂
- power(m,n)： 返回 m 的 n 次幂
- log(m,n)： 返回对数值
- ln(m)：返回 m（m>0）的自然对数（以常数 e 为底数的对数叫做自然对数，记作 ln N(N>0).）
- sqrt(m)：返回参数的平方根 由于负数是不能开平方的 ,所以 m 不能为负数

注意： 三角函数默认的参数认定为弧度制

对数字的处理，在财务系统或银行系统中用的最多，不同的处理方法，对财务报表有不同的结果。

1.3.1 trunc(n,[m])

1.3.1.1 对日期型数据截去取整

TRUNC:表示对日期进行截取

下面是该函数的使用情况：

```
SELECT trunc(SYSDATE),
       TRUNC(SYSDATE - 1),
       SYSDATE - 1,
       trunc(SYSDATE, 'dd'), --返回当前年月日
       trunc(SYSDATE, 'd'), --返回当前星期的第一天。
       trunc(SYSDATE, 'mm'), --返回当月第一天。
       trunc(SYSDATE, 'yyyy'), --返回当年第一天。
       trunc(sysdate,'mi'), --截取到分钟，即秒是以00来显示的，可以和 （SYSDATE - 1）的时间做比较
       TRUNC(TO_DATE('24-12月-1999 08:00 ','dd-mon-yyyy hh:mi AM')),
       TRUNC(TO_DATE('24-12月-1999 08:37 ','dd-mon-yyyy hh:mi AM'),'hh')
FROM   dual;
```

转换为列模式后的结果：

Row 1	Fields
TRUNC(SYSDATE)	2012/4/2
TRUNC(SYSDATE-1)	2012/4/1
SYSDATE-1	2012/4/1 9:28:48
TRUNC(SYSDATE,'DD')	2012/4/2
TRUNC(SYSDATE,'D')	2012/4/1
TRUNC(SYSDATE,'MM')	2012/4/1
TRUNC(SYSDATE,'YYYY')	2012/1/1
TRUNC(SYSDATE,'MI')	2012/4/2 9:28:00
TRUNC(TO_DATE('24-12月-199908:	1999/12/24
TRUNC(TO_DATE('24-12月-199908:	1999/12/24 8:00:00

1.3.1.2 对数值截去取整返回类型为数值

```
SELECT  TRUNC(456.873),TRUNC(456.873,2),trunc(456.873,-2),trunc(456.873,-3)  FROM dual ;
```

TRUNC(456.873)	TRUNC(456.873,2)	TRUNC(456.873,-2)	TRUNC(456.873,-3)
456	456.87	400	0

TRUNC（for number）

TRUNC 函数返回处理后的数值，其工作机制与 ROUND 函数极为类似，只是该函数不对指定小数前或后的部分做相应舍入选择处理，而统统截去。

下面是该函数的使用情况：

TRUNC（89.985，2）=89.98

TRUNC（89.985）=89

TRUNC（89.985，-1）=80

注意：第二个参数可以为负数，表示为小数点左边指定位数后面的部分截去，即均以 0 记。与取整类似，比如参数为 1 即取整到十分位，如果是-1，则是取整到十位，以此类推。

1.3.2 Round

1.3.2.1 日期

ROUND:对日期进行四舍五入

如果按月进行四舍五入，15 日算上半月，会舍去；16 日算下半月，会进位

如果按日进行四舍五入，星期三算上半周，星期四算下半周，周日不变

```
select round(to_date('15/2/2011','DD/MM/YYYY'),'month') from dual
Select ROUND(Sysdate,'YEAR') From DUAL
```

例如：Assume SYSDATE = '25-JUL-95':

ROUND(SYSDATE,'MONTH') 则 01-AUG-95

ROUND(SYSDATE ,'YEAR') 则 01-JAN-96

1.3.2.2 数值

```
SELECT
round(156.00,-2),round(156.00,-1),round(156.00,-3),trunc(round(156.00,-1),-1)
FROM dual ;
```

ROUND(156.00,-2)	ROUND(156.00,-1)	ROUND(156.00,-3)	TRUNC(ROUND(156.00,-1),-1)
200	160	0	160

1.4 日期函数

1. 日期函数用于处理 date 类型的数据。
2. 在日期上加上或减去一个数字结果仍为日期。
3. 两个日期相减返回日期之间相差的天数。
4. 可以用数字除 24 来向日期中加上或减去小时。
5. 默认情况下日期格式是 DD-MON-RR 即 12-7 月-78，这个可以用如下语句来查询：

```
select sys_context('userenv','nls_date_format') from dual;
```

既然数据库以数字方式存储日期，你就可以用算术运算符进行计算，例如，加或减。你可以加或减数字常数以及日期。从日期加或者减一个数，结果是一个日期值，两个日期相减，得到两个日期之间的天数，用小时数除以 24，可以加小时到日期上。

运算	结果	说明
date + number	日期	加一个天数到一个日期上
date - number	日期	从一个日期上减一个天数
date - date	天数	用一个日期减另一个日期
date + number/24	日期	加一个小时数到一个日期上

(1)sysdate: 该函数返回系统时间
(2)add_months(d,n) 该函数将给定的日期增加 n 个月
(3)last_day(d)：返回指定日期所在月份的最后一天

注意：
sysdate:年月日时分秒
日期+-1,都代表一天的时间,比如：

```
Select TRUNC(Sysdate-365) From DUAL
Select Sysdate-1/24/60 From DUAL
```

求月份：

```
SELECT to_char(SYSDATE, 'mm') FROM dual;
```

TO_CHAR(SYSDATE,'MM')
05

```
--上一个月
SELECT to_char(add_months(trunc(SYSDATE), -1), 'yyyy-mm') FROM dual;
```

TO_CHAR(ADD_MONTHS(TRUNC(SYSDA
2012-04

```
--下一个月
SELECT to_char(add_months(trunc(SYSDATE), 1), 'yyyy-mm') FROM dual;
```

TO_CHAR(ADD_MONTHS(TRUNC(SYSDA
2012-06

```
--去年
SELECT SYSDATE,
```

```
      add_months(SYSDATE, -12)
FROM   dual;
```

SYSDATE	ADD_MONTHS(SYSDATE,-12)
2012/5/13 16:39:36	2011/5/13 16:39:36

技巧：

如果某一个需求需要求在某一个月中的情况（比如：2012-1-1到2012-1-31），此时可以换种思维，就是用年份和月份来表示，将他们截取出来，如 createdate这个列就可以表示成：

```
to_char(createdate,'yyyy-mm')=to_char(sysdate,'yyyy-mm')
```

1.4.1 MONTHS_BETWEEN

MONTHS_BETWEEN:表示两个日期的月份之差，即在给定的两个日期之间有多少个月

```
select months_between(sysdate,'16-9月-2001') from dual;
Select EMPNO,HIREDATE,MONTHS_BETWEEN(Sysdate,HIREDATE)/12 From EMP;
 SELECT SYSDATE,
       MONTHS_BETWEEN(SYSDATE, TO_DATE('2016-12-20', 'YYYY-MM-DD')),
       MONTHS_BETWEEN(SYSDATE, TO_DATE('2016-10-20', 'YYYY-MM-DD'))
  FROM DUAL;
```

	SYSDATE	MONTHS_BETWEEN(SYSDATE,TO_DATE	MONTHS_BETWEEN(SYSDATE,TO_DATE
1	2016-11-20 19:23:26	-1	1

1.4.2 ADD_MONTHS(d,n)

ADD_MONTHS(d,n):当 n 为正数时，该函数将给定的日期增加 n 个月，为负数时减去 n 个月，该函数很常用，可以用来表示上个月、下个月，去年和下一年等等。

```
select add_months(sysdate,4) from dual
Select HIREDATE,ADD_MONTHS(HIREDATE,3) From EMP
SELECT SYSDATE, ADD_MONTHS(SYSDATE, 1), ADD_MONTHS(SYSDATE, -1) FROM DUAL;
```

	SYSDATE	ADD_MONTHS(SYSDATE,1)	ADD_MONTHS(SYSDATE,-1)
1	2016-11-20 19:27:58	2016-12-20 19:27:58	2016-10-20 19:27:58

ADD_MONTHS(x, y)用于计算 x 加上 y 个月的结果。如果 y 是负数，就从 x 中减去 y 个月。下面这个例子在 2007 年 1 月 1 日上加上 13 个月：

```
SELECT ADD_MONTHS('01-JAN-2007', 13)FROM dual;ADD_MONTH---------01-FEB-08
```

下面这个例子从 2008 年 1 月 1 日中减去 13 个月；注意本例实际上是使用 ADD_MONTHS 函数在这个日期上加上-13 个月：

```
SELECT ADD_MONTHS('01-JAN-2008', -13)FROM dual;ADD_MONTH---------01-DEC-06
```

ADD_MONTHS 函数可以用于时间和日期。例如，下面这个查询在时间值 2007 年 1 月 1 日下午 7 点 15 分 26 秒上增加两个月：

```
SELECT ADD_MONTHS(TO_DATE('01-JAN-2007 19:15:26','DD-MON-YYYY HH24:MI:SS'), 2)FROM
dual;ADD_MONTH---------01-MAR-07
```

下面这个查询重写了上面这个例子：它使用 TO_CHAR 函数将从 ADD_MONTHS 函数中返回的时间值转换为字符串，并指定格式为 DD-MON-YYYY HH24:MI:SS：

```
SELECT TO_CHAR(ADD_MONTHS(TO_DATE('01-JAN-2007 19:15:26','DD-MON-YYYY HH24:MI:SS'),
2), 'DD-MON-YYYY HH24:MI:SS')FROM dual;
```

```
TO_CHAR(ADD_MONTHS(T
--------------------
01-MAR-2007 19:15:26
```

1.4.3 NEXT_DAY(d,n)

NEXT_DAY(d,n)：返回以时间点 d 为基准(开始)，下一个"目标日 n"的日期

SELECT SYSDATE,NEXT_DAY(SYSDATE,'星期二') FROM DUAL;　表示下周二的日期

补充：修改当地语言：ALTER SESSION SET NLS_LANGUAGE='AMERICAN';
　　修改为中文：ALTER SESSION SET NLS_LANGUAGE='SIMPLIFIED CHINESE';

在英语的环境中，执行星期几时要用英文：

select next_day(sysdate,'Friday') from dual;

```
SQL> SELECT SYSDATE,NEXT_DAY(SYSDATE,'星期二') FROM DUAL;
SELECT SYSDATE,NEXT_DAY(SYSDATE,'星期二') FROM DUAL
                          *
ERROR at line 1:
ORA-01846: not a valid day of the week


SQL> ALTER SESSION SET NLS_LANGUAGE='SIMPLIFIED CHINESE';

会话已更改。

SQL>  SELECT SYSDATE,NEXT_DAY(SYSDATE,'星期二') FROM DUAL;

SYSDATE           NEXT_DAY(SYSDATE,'
------------------- -------------------
2016-11-20 19:50:07 2016-11-22 19:50:07

SQL> select next_day(sysdate,'Friday') from dual;
select next_day(sysdate,'Friday') from dual
                  *
第 1 行出现错误:
ORA-01846: 周中的日无效


SQL> ALTER SESSION SET NLS_LANGUAGE='AMERICAN';

Session altered.

SQL>  select next_day(sysdate,'Friday') from dual;

NEXT_DAY(SYSDATE,'F
-------------------
2016-11-25 19:50:55

SQL> select next_day(sysdate,1) from dual;

NEXT_DAY(SYSDATE,1)
-------------------
2016-11-27 19:52:50

SQL>  ALTER SESSION SET NLS_LANGUAGE='SIMPLIFIED CHINESE';
```

```
会话已更改。

SQL> select next_day(sysdate,1) from dual;

NEXT_DAY(SYSDATE,1)
-------------------
2016-11-27 19:52:59

SQL>
```

在生活中，为了避免因为语言问题导致的命令错误，常用下面的方法：

select next_day(sysdate,1) from dual; --1 表示星期日，7表示星期六，依此类推，**不过在用数字作为日期的时候,next_day函数不能够直接赋值给变量,只能够通过select into方式.**

```
SELECT to_char(next_day(last_day(trunc(SYSDATE)), 7) - 7, 'YYYY-MM-DD')
INTO   v_sdate
FROM   dual;

--- 或者 下边
v_sdate := to_char(next_day(last_day(trunc(SYSDATE)), '星期六') - 7,
                   'YYYY-MM-DD');
--这种错误，next_day 使用数字不能直接赋值 ，困扰我很久 2013/1/20 16:40 躺床上突然想到才解决
v_sdate := to_char(next_day(last_day(trunc(SYSDATE)), 7) - 7, 'YYYY-MM-DD');
```

```
-----  判断今天是否月份最后一天
 WITH T AS
 (SELECT   sysdate DAT FROM DUAL)
 SELECT DECODE(DAT, LAST_DAY(DAT), 'yes', 'no') FROM T;
```

--- 判断 '2014-04-30'是否所在月份最后一天

```
WITH T AS
(SELECT DATE '2014-04-30' DAT FROM DUAL)
SELECT DECODE(DAT, LAST_DAY(DAT), 'yes', 'no') FROM T;
```

1.4.4 LAST_DAY

LAST_DAY ：计算给定日期的最后一天，当月的最后一天

```
Select HIREDATE,LAST_DAY(HIREDATE)-HIREDATE From EMP;
SELECT last_day(TO_DATE('201302', 'yyyymm')),
       last_day(trunc(SYSDATE))
FROM   dual;
```

LAST_DAY(TO_DATE('201302','YYY	LAST_DAY(TRUNC(SYSDATE))
2013/2/28	2013/1/31

一、 next_day 和 last_day 联合使用

```
SELECT trunc(SYSDATE, 'MM') 本月第一天,
       add_months(trunc(SYSDATE, 'MM'), 1) 下个月第一天,
       add_months(trunc(SYSDATE, 'MM'), 1) - 1 本月最后一天,
       next_day(add_months(trunc(SYSDATE, 'MM'), 1) - 1, 7) 本月最后一天的下一个周六,
       next_day(add_months(trunc(SYSDATE, 'MM'), 1) - 1, 7) - 7 本月最后一个周六,
       next_day(last_day(trunc(SYSDATE)), 7) - 7 本月最后一个周六
FROM   dual;
```

本月第一天	下个月第一天	本月最后一天	本月最后一天的下一个周六	本月最后一个周六	本月最后一个周六
2013/1/1	2013/2/1	2013/1/31	2013/2/2	2013/1/26	2013/1/26

1.4.5 NEW_TIME

NEW_TIME: 调整时区

下边给出了所有的时区

简写	时区
AST or ADT	大西洋标准时间
HST or HDT	阿拉斯加_夏威夷时间
BST or BDT	英国夏令时
MST or MDT	美国山区时间
CST or CDT	美国中央时区
NST	新大陆标准时间
EST or EDT	美国东部时区
PST or PDT	太平洋标准时间
GMT	格伦威治标准时间
YST or YDT	Yukon 标准时间

1.4.6 求 2 个日期之间的时间差

在 ORACLE 里日期类型是可以直接进行比较的。举个例子

```
SQL> SELECT to_date('2006-12-05', 'yyyy-mm-dd') - to_date('2006-12-07', 'yyyy-mm-dd')
FROM dual ;

TO_DATE('2006-12-05','YYYY-MM-DD')-TO_DATE('2006-12-07','YYYY-MM-DD')
---------------------------------------------------------------------
                                                                   -2
```

这说明 2006-12-05 要比 7 号早 2 天。

```
SQL> SELECT to_date('2006-12-07 14:23:24' , 'yyyy-mm-dd hh24:mi:ss') + 1/24 FROM dual ;

TO_DATE('2006-12-07
-------------------
2006-12-07 15:23:24
```

SQL>

这是说明在某一时间上加一小时，1 是代表一天，1/24 就是一小时，同理 1/24/60 就是一分钟

select trunc(sysdate) - trunc(to_date('2006-05-28 10:20','YYYY-MM-DD HH24:MI')) from dual

Oracle 中计算时间差是经常用到的。可以使用"日期 1－日期 2"并加以运算，来获得你要想的时间差：天、小时、分钟或者秒。

例如：

```
SELECT TO_DATE('2012-02-20 17:45:04', 'yyyy-mm-dd hh24:mi:ss') -
       TO_DATE('2012-02-19 08:34:04', 'yyyy-mm-dd hh24:mi:ss') AS DAY
FROM   dual;
```

结果：

	DAY
1	1.38263888888889

这里的 TO_DATE 很有用，它决定你的时间存储格式。

那么如果要获取相应的时间单位，下面：

一、 以天为单位

round(to_number(end-date-start_date))

例如：

```
SELECT round(to_number(TO_DATE('2012-02-20 17:45:04',
                               'yyyy-mm-dd hh24:mi:ss') -
                       TO_DATE('2012-02-19 08:34:04',
                               'yyyy-mm-dd hh24:mi:ss'))) AS DAY
FROM   dual;
```

结果：

	DAY
1	1

二、 以小时为单位

round(to_number(end-date-start_date)*24)

例如：

```
SELECT round(to_number(TO_DATE('2012-02-20 17:45:04',
                               'yyyy-mm-dd hh24:mi:ss') -
                       TO_DATE('2012-02-19 08:34:04',
                               'yyyy-mm-dd hh24:mi:ss')) * 24) AS Hour
FROM   dual;
```

结果：

	HOUR
1	33

三、 以分钟为单位

round(to_number(end-date-start_date)*1440)

例如：

```
SELECT round(to_number(TO_DATE('2012-02-20 17:45:04',
                               'yyyy-mm-dd hh24:mi:ss') -
                       TO_DATE('2012-02-19 08:34:04',
                               'yyyy-mm-dd hh24:mi:ss')) * 1440) AS Minite
FROM   dual;
```

结果：

	MINITE
1	1991

四、 秒

```
ROUND(TO_NUMBER(END_DATE - START_DATE) * 24 * 60 * 60)
```

五、 毫秒：

```
ROUND(TO_NUMBER(END_DATE - START_DATE) * 24 * 60 * 60 * 1000)
```

1.4.7 常用之日期格式

```
日期格式                    說明
-------------------------------------------------------------------------
YYYY/MM/DD                  -- 年/月/日
YYYY                        -- 年(4 位)
YYY                         -- 年(3 位)
YY                          -- 年(2 位)
MM                          -- 月份
DD                          -- 日期
D                           -- 星期
                               -- 星期日 = 1  星期一 = 2 星期二 = 3
                            -- 星期三 = 4  星期四 = 5 星期五 = 6 星期六 = 7

DDD                         -- 一年之第幾天
WW                          -- 一年之第幾週
W                           -- 一月之第幾週
YYYY/MM/DD HH24:MI:SS       -- 年/月/日 時(24 小時制):分:秒
YYYY/MM/DD HH:MI:SS         -- 年/月/日 時(非 24 小時制):分:秒
J                    -- Julian day, julian date 指的是公元前 4712 年 1 月 1 日起经过的天
数.,104235，1 指 21 世纪，04 指第四年，235 指这年的第几天!
RR/MM/DD                    -- 公元 2000 問題
```

```
-- 00-49 = 下世紀；50-99 = 本世紀
    ex.
    select to_char(sysdate,''YYYY/MM/DD'') FROM DUAL;          -- 2007/09/20
    select to_char(sysdate,''YYYY'') FROM DUAL;                -- 2007
    select to_char(sysdate,''YYY'') FROM DUAL;                 -- 007
    select to_char(sysdate,''YY'') FROM DUAL;                  -- 07
    select to_char(sysdate,''MM'') FROM DUAL;                  -- 09
    select to_char(sysdate,''DD'') FROM DUAL;                  -- 20
    select to_char(sysdate,''D'') FROM DUAL;                   -- 5
    select to_char(sysdate,''DDD'') FROM DUAL;                 -- 263
    select to_char(sysdate,''WW'') FROM DUAL;                  -- 38
    select to_char(sysdate,''W'') FROM DUAL;                   -- 3
    select to_char(sysdate,''YYYY/MM/DD HH24:MI:SS'') FROM DUAL;  -- 2007/09/20 15:24:13
    select to_char(sysdate,''YYYY/MM/DD HH:MI:SS'') FROM DUAL;    -- 2007/09/20 03:25:23
    select to_char(sysdate,''J'') FROM DUAL;                   -- 2454364
   select to_char(sysdate-365,'j') from dual;
   select to_date (to_char(sysdate-365,'j') + trunc(dbms_random.value(0,365)),'j')
from dual;
    select to_char(sysdate,''RR/MM/DD'') FROM DUAL;            -- 07/09/20
```

1.4.8 时区

如何判断数据库的时区？

```
select dbtimezone from dual;
select sessiontimezone from dual;
select systimestamp from dual;
select current_date from dual;
select current_timestamp from dual;
```

1.5 常用日期查询

1.获取当前月份的第一天

运行这个命令能快速返回当前月份的第一天。你可以用任何的日期值替换“SYSDATE”来指定查询的日期。

```
SELECT TRUNC (SYSDATE, 'MONTH') "First day of current month"
FROM DUAL;
```

2.获取当前月份的最后一天

这个查询语句类似于上面那个语句，而且充分照顾到了闰年，所以当二月份有 29 号，那么就会返回 29/2 。你可以用任何的日期值替换“SYSDATE”来指定查询的日期。

```
SELECT TRUNC (LAST_DAY (SYSDATE)) "Last day of current month"
FROM DUAL;
```

3.获取当前年份的第一天

每年的第一天都是 1 月 1 日，这个查询语句可以使用在存储过程中，需要对当前年份第一天做一些计算的时候。你可以用任何的日期值替换“SYSDATE”来指定查询的日期。

```
SELECT TRUNC (SYSDATE, 'YEAR') "Year First Day" FROM DUAL;
```

4.获取当前年份的最后一天

类似于上面的查询语句。你可以用任何的日期值替换“SYSDATE”来指定查询的日期。

```
SELECT ADD_MONTHS (TRUNC (SYSDATE, 'YEAR'), 12) - 1 "Year Last Day" FROM DUAL
```

5.获取当前月份的天数

这个语句非常有用，可以计算出当前月份的天数。你可以用任何的日期值替换“SYSDATE”来指定查询的日期。

```
SELECT CAST (TO_CHAR (LAST_DAY (SYSDATE), 'dd') AS INT) number_of_days
FROM DUAL;
```

6.获取当前月份剩下的天数

下面的语句用来计算当前月份剩下的天数。你可以用任何的日期值替换“SYSDATE”来指定查询的日期。

```
SELECT SYSDATE,
LAST_DAY (SYSDATE) "Last",
LAST_DAY (SYSDATE) - SYSDATE "Days left"
FROM DUAL;
```

7.获取两个日期之间的天数

使用这个语句来获取两个不同日期自检的天数。

```
SELECT ROUND ( (MONTHS_BETWEEN ('01-Feb-2014', '01-Mar-2012') * 30), 0)
   num_of_days
FROM DUAL;

OR

SELECT TRUNC(sysdate) - TRUNC(e.hire_date) FROM employees;
```

如果你需要查询一些特定日期的天数，可以使用第二个查询语句。这个例子是计算员工入职的天数。

8.显示当前年份截止到上个月每个月份开始和结束的日期

这个是个很聪明的查询语句，用来显示当前年份每个月的开始和结束的日期，你可以使用这个进行一些类型的计算。你可以用任何的日期值替换“SYSDATE”来指定查询的日期。

```
SELECT ADD_MONTHS (TRUNC (SYSDATE, 'MONTH'), i) start_date,
      TRUNC (LAST_DAY (ADD_MONTHS (SYSDATE, i))) end_date
  FROM XMLTABLE (
         'for $i in 0 to xs:int(D) return $i'
         PASSING XMLELEMENT (
                  d,
                  FLOOR (
                     MONTHS_BETWEEN (
                        ADD_MONTHS (TRUNC (SYSDATE, 'YEAR') - 1, 12),
                        SYSDATE)))
         COLUMNS i INTEGER PATH '.');
```

9.获取直到目前为止今天过去的秒数（从 00：00 开始算）

```
SELECT (SYSDATE - TRUNC (SYSDATE)) * 24 * 60 * 60 num_of_sec_since_morning
FROM DUAL;
```

10.获取今天剩下的秒数（直到 23：59：59 结束）

```
SELECT (TRUNC (SYSDATE+1) - SYSDATE) * 24 * 60 * 60 num_of_sec_left
  FROM DUAL;
```

About Me

- 本文作者：小麦苗，只专注于数据库的技术，更注重技术的运用
- 本文在 itpub（http://blog.itpub.net/26736162）、博客园（http://www.cnblogs.com/lhrbest）和个人微信公众号（xiaoma
- 本文 itpub 地址：http://blog.itpub.net/26736162/viewspace-2128764/
- 本文博客园地址：http://www.cnblogs.com/lhrbest/p/6083478.html
- 本文 pdf 版及小麦苗云盘地址：http://blog.itpub.net/26736162/viewspace-1624453/
- QQ 群：230161599　　微信群：私聊
- 联系我请加 QQ 好友(642808185)，注明添加缘由
- 于 2016-11-20 18:00 ~ 2016-11-20 21:00 在泰兴公寓完成
- 文章内容来源于小麦苗的学习笔记，部分整理自网络，若有侵权或不当之处还请谅解

- 版权所有，欢迎分享本文，转载请保留出处

手机长按下图识别二维码或微信客户端扫描下边的二维码来关注小麦苗的微信公众号：xiaomaimiaolhr，免费学习最实用的数据库技术。