mysql中sql语句进行日期比较

这里是一个使用日期函数的例子。下面的查询选择了所有记录，其date\_col的值是在最后30天以内：

mysql> SELECT something FROM table  
WHERE TO\_DAYS(NOW()) - TO\_DAYS(date\_col) <= 30;

DAYOFWEEK(date)  
返回日期date的星期索引(1=星期天，2=星期一, ……7=星期六)。这些索引值对应于ODBC标准。  
mysql> select DAYOFWEEK('1998-02-03');  
-> 3

WEEKDAY(date)  
返回date的星期索引(0=星期一，1=星期二, ……6= 星期天)。  
mysql> select WEEKDAY('1997-10-04 22:23:00');  
-> 5  
mysql> select WEEKDAY('1997-11-05');  
-> 2

DAYOFMONTH(date)  
返回date的月份中日期，在1到31范围内。  
mysql> select DAYOFMONTH('1998-02-03');  
-> 3

DAYOFYEAR(date)  
返回date在一年中的日数, 在1到366范围内。  
mysql> select DAYOFYEAR('1998-02-03');  
-> 34

MONTH(date)  
返回date的月份，范围1到12。  
mysql> select MONTH('1998-02-03');  
-> 2

DAYNAME(date)  
返回date的星期名字。  
mysql> select DAYNAME("1998-02-05");  
-> 'Thursday'

MONTHNAME(date)  
返回date的月份名字。  
mysql> select MONTHNAME("1998-02-05");  
-> 'February'

QUARTER(date)  
返回date一年中的季度，范围1到4。  
mysql> select QUARTER('98-04-01');  
-> 2

WEEK(date)

WEEK(date,first)  
对于星期天是一周的第一天的地方，有一个单个参数，返回date的周数，范围在0到52。2个参数形式WEEK()允许  
你指定星期是否开始于星期天或星期一。如果第二个参数是0，星期从星期天开始，如果第二个参数是1，  
从星期一开始。  
mysql> select WEEK('1998-02-20');  
-> 7  
mysql> select WEEK('1998-02-20',0);  
-> 7  
mysql> select WEEK('1998-02-20',1);  
-> 8

YEAR(date)  
返回date的年份，范围在1000到9999。  
mysql> select YEAR('98-02-03');  
-> 1998

HOUR(time)  
返回time的小时，范围是0到23。  
mysql> select HOUR('10:05:03');  
-> 10

MINUTE(time)  
返回time的分钟，范围是0到59。  
mysql> select MINUTE('98-02-03 10:05:03');  
-> 5

SECOND(time)  
回来time的秒数，范围是0到59。  
mysql> select SECOND('10:05:03');  
-> 3

PERIOD\_ADD(P,N)  
增加N个月到阶段P（以格式YYMM或YYYYMM)。以格式YYYYMM返回值。注意阶段参数P不是日期值。  
mysql> select PERIOD\_ADD(9801,2);  
-> 199803

PERIOD\_DIFF(P1,P2)  
返回在时期P1和P2之间月数，P1和P2应该以格式YYMM或YYYYMM。注意，时期参数P1和P2不是日期值。  
mysql> select PERIOD\_DIFF(9802,199703);  
-> 11

DATE\_ADD(date,INTERVAL expr type)

DATE\_SUB(date,INTERVAL expr type)

ADDDATE(date,INTERVAL expr type)

SUBDATE(date,INTERVAL expr type)  
这些功能执行日期运算。对于MySQL 3.22，他们是新的。ADDDATE()和SUBDATE()是DATE\_ADD()和DATE\_SUB()的同义词。  
在MySQL 3.23中，你可以使用+和-而不是DATE\_ADD()和DATE\_SUB()。（见例子）date是一个指定开始日期的  
DATETIME或DATE值，expr是指定加到开始日期或从开始日期减去的间隔值一个表达式，expr是一个字符串；它可以以  
一个“-”开始表示负间隔。type是一个关键词，指明表达式应该如何被解释。EXTRACT(type FROM date)函数从日期  
中返回“type”间隔。下表显示了type和expr参数怎样被关联： type值 含义 期望的expr格式  
SECOND 秒 SECONDS  
MINUTE 分钟 MINUTES  
HOUR 时间 HOURS  
DAY 天 DAYS  
MONTH 月 MONTHS  
YEAR 年 YEARS  
MINUTE\_SECOND 分钟和秒 "MINUTES:SECONDS"  
HOUR\_MINUTE 小时和分钟 "HOURS:MINUTES"  
DAY\_HOUR 天和小时 "DAYS HOURS"  
YEAR\_MONTH 年和月 "YEARS-MONTHS"  
HOUR\_SECOND 小时, 分钟， "HOURS:MINUTES:SECONDS"  
DAY\_MINUTE 天, 小时, 分钟 "DAYS HOURS:MINUTES"  
DAY\_SECOND 天, 小时, 分钟, 秒 "DAYS HOURS:MINUTES:SECONDS"

MySQL在expr格式中允许任何标点分隔符。表示显示的是建议的分隔符。如果date参数是一个DATE值并且你的计算仅仅  
包含YEAR、MONTH和DAY部分(即，没有时间部分)，结果是一个DATE值。否则结果是一个DATETIME值。

mysql> SELECT "1997-12-31 23:59:59" + INTERVAL 1 SECOND;  
-> 1998-01-01 00:00:00  
mysql> SELECT INTERVAL 1 DAY + "1997-12-31";  
-> 1998-01-01  
mysql> SELECT "1998-01-01" - INTERVAL 1 SECOND;  
-> 1997-12-31 23:59:59  
mysql> SELECT DATE\_ADD("1997-12-31 23:59:59",  
INTERVAL 1 SECOND);  
-> 1998-01-01 00:00:00  
mysql> SELECT DATE\_ADD("1997-12-31 23:59:59",  
INTERVAL 1 DAY);  
-> 1998-01-01 23:59:59  
mysql> SELECT DATE\_ADD("1997-12-31 23:59:59",  
INTERVAL "1:1" MINUTE\_SECOND);  
-> 1998-01-01 00:01:00  
mysql> SELECT DATE\_SUB("1998-01-01 00:00:00",  
INTERVAL "1 1:1:1" DAY\_SECOND);  
-> 1997-12-30 22:58:59  
mysql> SELECT DATE\_ADD("1998-01-01 00:00:00",  
INTERVAL "-1 10" DAY\_HOUR);  
-> 1997-12-30 14:00:00  
mysql> SELECT DATE\_SUB("1998-01-02", INTERVAL 31 DAY);  
-> 1997-12-02  
mysql> SELECT EXTRACT(YEAR FROM "1999-07-02");  
-> 1999  
mysql> SELECT EXTRACT(YEAR\_MONTH FROM "1999-07-02 01:02:03");  
-> 199907  
mysql> SELECT EXTRACT(DAY\_MINUTE FROM "1999-07-02 01:02:03");  
-> 20102

如果你指定太短的间隔值(不包括type关键词期望的间隔部分)，MySQL假设你省掉了间隔值的最左面部分。例如，  
如果你指定一个type是DAY\_SECOND，值expr被希望有天、小时、分钟和秒部分。如果你象"1:10"这样指定值，  
MySQL假设日子和小时部分是丢失的并且值代表分钟和秒。换句话说，"1:10" DAY\_SECOND以它等价于"1:10" MINUTE\_SECOND  
的方式解释，这对那MySQL解释TIME值表示经过的时间而非作为一天的时间的方式有二义性。如果你使用确实不正确的日期，  
结果是NULL。如果你增加MONTH、YEAR\_MONTH或YEAR并且结果日期大于新月份的最大值天数，日子在新月用最大的天调整。

mysql> select DATE\_ADD('1998-01-30', Interval 1 month);  
-> 1998-02-28

注意，从前面的例子中词INTERVAL和type关键词不是区分大小写的。  
TO\_DAYS(date)  
给出一个日期date，返回一个天数(从0年的天数)。  
mysql> select TO\_DAYS(950501);  
-> 728779  
mysql> select TO\_DAYS('1997-10-07');  
-> 729669

TO\_DAYS()不打算用于使用格列高里历(1582)出现前的值。

FROM\_DAYS(N)  
给出一个天数N，返回一个DATE值。  
mysql> select FROM\_DAYS(729669);  
-> '1997-10-07'

TO\_DAYS()不打算用于使用格列高里历(1582)出现前的值。

DATE\_FORMAT(date,format)  
根据format字符串格式化date值。下列修饰符可以被用在format字符串中： %M 月名字(January……December)  
%W 星期名字(Sunday……Saturday)  
%D 有英语前缀的月份的日期(1st, 2nd, 3rd, 等等。）  
%Y 年, 数字, 4 位  
%y 年, 数字, 2 位  
%a 缩写的星期名字(Sun……Sat)  
%d 月份中的天数, 数字(00……31)  
%e 月份中的天数, 数字(0……31)  
%m 月, 数字(01……12)  
%c 月, 数字(1……12)  
%b 缩写的月份名字(Jan……Dec)  
%j 一年中的天数(001……366)  
%H 小时(00……23)  
%k 小时(0……23)  
%h 小时(01……12)  
%I 小时(01……12)  
%l 小时(1……12)  
%i 分钟, 数字(00……59)  
%r 时间,12 小时(hh:mm:ss [AP]M)  
%T 时间,24 小时(hh:mm:ss)  
%S 秒(00……59)  
%s 秒(00……59)  
%p AM或PM  
%w 一个星期中的天数(0=Sunday ……6=Saturday ）  
%U 星期(0……52), 这里星期天是星期的第一天  
%u 星期(0……52), 这里星期一是星期的第一天  
%% 一个文字“%”。

所有的其他字符不做解释被复制到结果中。

mysql> select DATE\_FORMAT('1997-10-04 22:23:00', '%W %M %Y');  
-> 'Saturday October 1997'  
mysql> select DATE\_FORMAT('1997-10-04 22:23:00', '%H:%i:%s');  
-> '22:23:00'  
mysql> select DATE\_FORMAT('1997-10-04 22:23:00',  
'%D %y %a %d %m %b %j');  
-> '4th 97 Sat 04 10 Oct 277'  
mysql> select DATE\_FORMAT('1997-10-04 22:23:00',  
'%H %k %I %r %T %S %w');  
-> '22 22 10 10:23:00 PM 22:23:00 00 6'  
MySQL3.23中，在格式修饰符字符前需要%。在MySQL更早的版本中，%是可选的。

TIME\_FORMAT(time,format)  
这象上面的DATE\_FORMAT()函数一样使用，但是format字符串只能包含处理小时、分钟和秒的那些格式修饰符。  
其他修饰符产生一个NULL值或0。  
CURDATE()

CURRENT\_DATE  
以'YYYY-MM-DD'或YYYYMMDD格式返回今天日期值，取决于函数是在一个字符串还是数字上下文被使用。  
mysql> select CURDATE();  
-> '1997-12-15'  
mysql> select CURDATE() + 0;  
-> 19971215

CURTIME()

CURRENT\_TIME  
以'HH:MM:SS'或HHMMSS格式返回当前时间值，取决于函数是在一个字符串还是在数字的上下文被使用。  
mysql> select CURTIME();  
-> '23:50:26'  
mysql> select CURTIME() + 0;  
-> 235026

NOW()

SYSDATE()

CURRENT\_TIMESTAMP  
以'YYYY-MM-DD HH:MM:SS'或YYYYMMDDHHMMSS格式返回当前的日期和时间，取决于函数是在一个字符串还是在数字的  
上下文被使用。  
mysql> select NOW();  
-> '1997-12-15 23:50:26'  
mysql> select NOW() + 0;  
-> 19971215235026

UNIX\_TIMESTAMP()

UNIX\_TIMESTAMP(date)  
如果没有参数调用，返回一个Unix时间戳记(从'1970-01-01 00:00:00'GMT开始的秒数)。如果UNIX\_TIMESTAMP()用一  
个date参数被调用，它返回从'1970-01-01 00:00:00' GMT开始的秒数值。date可以是一个DATE字符串、一个DATETIME  
字符串、一个TIMESTAMP或以YYMMDD或YYYYMMDD格式的本地时间的一个数字。  
mysql> select UNIX\_TIMESTAMP();  
-> 882226357  
mysql> select UNIX\_TIMESTAMP('1997-10-04 22:23:00');  
-> 875996580

当UNIX\_TIMESTAMP被用于一个TIMESTAMP列，函数将直接接受值，没有隐含的“string-to-unix-timestamp”变换。

FROM\_UNIXTIME(unix\_timestamp)  
以'YYYY-MM-DD HH:MM:SS'或YYYYMMDDHHMMSS格式返回unix\_timestamp参数所表示的值，取决于函数是在一个字符串  
还是或数字上下文中被使用。  
mysql> select FROM\_UNIXTIME(875996580);  
-> '1997-10-04 22:23:00'  
mysql> select FROM\_UNIXTIME(875996580) + 0;  
-> 19971004222300

FROM\_UNIXTIME(unix\_timestamp,format)  
返回表示 Unix 时间标记的一个字符串，根据format字符串格式化。format可以包含与DATE\_FORMAT()函数列出的条  
目同样的修饰符。  
mysql> select FROM\_UNIXTIME(UNIX\_TIMESTAMP(),  
'%Y %D %M %h:%i:%s %x');  
-> '1997 23rd December 03:43:30 x'

SEC\_TO\_TIME(seconds)  
返回seconds参数，变换成小时、分钟和秒，值以'HH:MM:SS'或HHMMSS格式化，取决于函数是在一个字符串还是在数字  
上下文中被使用。  
mysql> select SEC\_TO\_TIME(2378);  
-> '00:39:38'  
mysql> select SEC\_TO\_TIME(2378) + 0;  
-> 3938

TIME\_TO\_SEC(time)  
返回time参数，转换成秒。  
mysql> select TIME\_TO\_SEC('22:23:00');  
-> 80580  
mysql> select TIME\_TO\_SEC('00:39:38');  
-> 2378  
DATE\_FORMAT(date,format)  
根据format字符串格式化date值。下列修饰符可以被用在format字符串中： %M 月名字(January……December)  
%W 星期名字(Sunday……Saturday)  
%D 有英语前缀的月份的日期(1st, 2nd, 3rd, 等等。）  
%Y 年, 数字, 4 位  
%y 年, 数字, 2 位  
%a 缩写的星期名字(Sun……Sat)  
%d 月份中的天数, 数字(00……31)  
%e 月份中的天数, 数字(0……31)  
%m 月, 数字(01……12)  
%c 月, 数字(1……12)  
%b 缩写的月份名字(Jan……Dec)  
%j 一年中的天数(001……366)  
%H 小时(00……23)  
%k 小时(0……23)  
%h 小时(01……12)  
%I 小时(01……12)  
%l 小时(1……12)  
%i 分钟, 数字(00……59)  
%r 时间,12 小时(hh:mm:ss [AP]M)  
%T 时间,24 小时(hh:mm:ss)  
%S 秒(00……59)  
%s 秒(00……59)  
%p AM或PM  
%w 一个星期中的天数(0=Sunday ……6=Saturday ）  
%U 星期(0……52), 这里星期天是星期的第一天  
%u 星期(0……52), 这里星期一是星期的第一天  
%% 一个文字“%”。 所有的其他字符不做解释被复制到结果中。

mysql> select DATE\_FORMAT('1997-10-04 22:23:00', '%W %M %Y');  
-> 'Saturday October 1997'  
mysql> select DATE\_FORMAT('1997-10-04 22:23:00', '%H:%i:%s');  
-> '22:23:00'  
mysql> select DATE\_FORMAT('1997-10-04 22:23:00',  
'%D %y %a %d %m %b %j');  
-> '4th 97 Sat 04 10 Oct 277'  
mysql> select DATE\_FORMAT('1997-10-04 22:23:00',  
'%H %k %I %r %T %S %w');  
-> '22 22 10 10:23:00 PM 22:23:00 00 6'  
MySQL3.23中，在格式修饰符字符前需要%。在MySQL更早的版本中，%是可选的。

TIME\_FORMAT(time,format)  
这象上面的DATE\_FORMAT()函数一样使用，但是format字符串只能包含处理小时、分钟和秒的那些格式修饰符。

其他修饰符产生一个NULL值或0。  
CURDATE()

CURRENT\_DATE  
以'YYYY-MM-DD'或YYYYMMDD格式返回今天日期值，取决于函数是在一个字符串还是数字上下文被使用。  
mysql> select CURDATE();  
-> '1997-12-15'  
mysql> select CURDATE() + 0;  
-> 19971215

CURTIME()

CURRENT\_TIME  
以'HH:MM:SS'或HHMMSS格式返回当前时间值，取决于函数是在一个字符串还是在数字的上下文被使用。  
mysql> select CURTIME();  
-> '23:50:26'  
mysql> select CURTIME() + 0;  
-> 235026

NOW()

SYSDATE()

CURRENT\_TIMESTAMP  
以'YYYY-MM-DD HH:MM:SS'或YYYYMMDDHHMMSS格式返回当前的日期和时间，取决于函数是在一个字符串还是在数字的  
上下文被使用。  
mysql> select NOW();  
-> '1997-12-15 23:50:26'  
mysql> select NOW() + 0;  
-> 19971215235026

UNIX\_TIMESTAMP()

UNIX\_TIMESTAMP(date)  
如果没有参数调用，返回一个Unix时间戳记(从'1970-01-01 00:00:00'GMT开始的秒数)。如果UNIX\_TIMESTAMP()用一个date参数被调用，它返回从'1970-01-01 00:00:00' GMT开始的秒数值。date可以是一个DATE字符串、一个DATETIME字符串、一个TIMESTAMP或以YYMMDD或YYYYMMDD格式的 本地时间的一个数字。  
mysql> select UNIX\_TIMESTAMP();  
-> 882226357  
mysql> select UNIX\_TIMESTAMP('1997-10-04 22:23:00');  
-> 875996580

当UNIX\_TIMESTAMP被用于一个TIMESTAMP列，函数将直接接受值，没有隐含的“string-to-unix-timestamp”变换 [www.knowsky.com](http://www.knowsky.com/)。

FROM\_UNIXTIME(unix\_timestamp)  
以'YYYY-MM-DD HH:MM:SS'或YYYYMMDDHHMMSS格式返回unix\_timestamp参数所表示的值，取决于函数是在一个字符串还是或数字上下文中被使用。  
mysql> select FROM\_UNIXTIME(875996580);  
-> '1997-10-04 22:23:00'  
mysql> select FROM\_UNIXTIME(875996580) + 0;  
-> 19971004222300

FROM\_UNIXTIME(unix\_timestamp,format)  
返回表示 Unix 时间标记的一个字符串，根据format字符串格式化。format可以包含与DATE\_FORMAT()函数列出的条目同样的修饰符。  
mysql> select FROM\_UNIXTIME(UNIX\_TIMESTAMP(),  
'%Y %D %M %h:%i:%s %x');  
-> '1997 23rd December 03:43:30 x'

SEC\_TO\_TIME(seconds)  
返回seconds参数，变换成小时、分钟和秒，值以'HH:MM:SS'或HHMMSS格式化，取决于函数是在一个字符串还是在数字上下文中被使用。  
mysql> select SEC\_TO\_TIME(2378);  
-> '00:39:38'  
mysql> select SEC\_TO\_TIME(2378) + 0;  
-> 3938

TIME\_TO\_SEC(time)  
返回time参数，转换成秒。  
mysql> select TIME\_TO\_SEC('22:23:00');  
-> 80580  
mysql> select TIME\_TO\_SEC('00:39:38');  
-> 2378