

MySQL函数

MySQL数据库提供了很多函数包括：

- 数学函数；
- 字符串函数；
- 日期和时间函数；
- 条件判断函数；
- 系统信息函数；
- 加密函数；
- 格式化函数；

一、数学函数

数学函数主要用于处理数字，包括整型、浮点数等。

函数	作用
ABS(x)	返回x的绝对值 SELECT ABS(-1) -- 返回1
CEIL(x),CEILING(x)	返回大于或等于x的最小整数 SELECT CEIL(1.5) -- 返回2
FLOOR(x)	返回小于或等于x的最大整数 SELECT FLOOR(1.5) -- 返回1
RAND()	返回0->1的随机数 SELECT RAND() --0.93099315644334
RAND(x)	返回0->1的随机数，x值相同时返回的随机数相同 SELECT RAND(2) --1.5865798029924
SIGN(x)	返回x的符号，x是负数、0、正数分别返回-1、0和1 SELECT SIGN(-10) -- (-1)
PI()	返回圆周率(3.141593) SELECT PI() --3.141593
TRUNCATE(x,y)	返回数值x保留到小数点后y位的值（与ROUND最大的区别是不会进行四舍五入） SELECT TRUNCATE(1.23456,3) -- 1.234
ROUND(x)	返回离x最近的整数 SELECT ROUND(1.23456) --1

ROUND(x,y)	保留x小数点后y位的值，但截断时要进行四舍五入 SELECT ROUND(1.23456,3) -- 1.235
POW(x,y).POWER(x,y)	返回x的y次方 SELECT POW(2,3) -- 8
SQRT(x)	返回x的平方根 SELECT SQRT(25) -- 5
EXP(x)	返回e的x次方 SELECT EXP(3) -- 20.085536923188
MOD(x,y)	返回x除以y以后的余数 SELECT MOD(5,2) -- 1
LOG(x)	返回自然对数(以e为底的对数) SELECT LOG(20.085536923188) -- 3
LOG10(x)	返回以10为底的对数 SELECT LOG10(100) -- 2
RADIANS(x)	将角度转换为弧度 SELECT RADIANS(180) -- 3.1415926535898
DEGREES(x)	将弧度转换为角度 SELECT DEGREES(3.1415926535898) -- 180
SIN(x)	求正弦值(参数是弧度) SELECT SIN(RADIANS(30)) -- 0.5
ASIN(x)	求反正弦值(参数是弧度)
COS(x)	求余弦值(参数是弧度)
ACOS(x)	求反余弦值(参数是弧度)
TAN(x)	求正切值(参数是弧度)
ATAN(x) ATAN2(x)	求反正切值(参数是弧度)
COT(x)	求余切值(参数是弧度)

二、字符串函数

字符串函数是MySQL中最常用的一类函数，字符串函数主要用于处理表中的字符串。

函数	说明
CHAR_LENGTH(s)	返回字符串s的字符数 SELECT CHAR_LENGTH('你好123') -- 5
LENGTH(s)	返回字符串s的长度 SELECT LENGTH('你好123') -- 9
CONCAT(s1,s2,...)	将字符串s1,s2等多个字符串合并为一个字符串 SELECT CONCAT('12','34') -- 1234
CONCAT_WS(x,s1,s2,...)	同CONCAT(s1,s2,...)函数，但是每个字符串直接要加上x SELECT CONCAT_WS('@','12','34') -- 12@34
INSERT(s1,x,len,s2)	将字符串s2替换s1的x位置开始长度为len的字符串 SELECT INSERT('12345',1,3,'abc') -- abc45
UPPER(s), UCASE(s)	将字符串s的所有字母变成大写字母 SELECT UPPER('abc') -- ABC
LOWER(s), LCASE(s)	将字符串s的所有字母变成小写字母 SELECT LOWER('ABC') -- abc
LEFT(s,n)	返回字符串s的前n个字符 SELECT LEFT('abcde',2) -- ab
RIGHT(s,n)	返回字符串s的后n个字符 SELECT RIGHT('abcde',2) -- de
LPAD(s1,len,s2)	字符串s2来填充s1的开始处，使字符串长度达到len SELECT LPAD('abc',5,'xx') -- xxabc
RPAD(s1,len,s2)	字符串s2来填充s1的结尾处，使字符串的长度达到len SELECT RPAD('abc',5,'xx') -- abcxx
LTRIM(s)	去掉字符串s开始处的空格
RTRIM(s)	去掉字符串s结尾处的空格

TRIM(s)	去掉字符串s开始和结尾处的空格
TRIM(s1 FROM s)	去掉字符串s中开始处和结尾处的字符串s1 SELECT TRIM('@' FROM '@@abc@@') -- abc
REPEAT(s, n)	将字符串s重复n次 SELECT REPEAT('ab',3) -- ababab
SPACE(n)	返回n个空格
REPLACE(s,s1,s2)	将字符串s2替代字符串s中的字符串s1 SELECT REPLACE('abc','a','x') --xbc
STRCMP(s1,s2)	比较字符串s1和s2
SUBSTRING(s,n,len)	获取从字符串s中的第n个位置开始长度为len的字符串
MID(s,n,len)	同SUBSTRING(s,n,len)
LOCATE(s1,s),POSITION(s1 IN s)	从字符串s中获取s1的开始位置 SELECT LOCATE('b', 'abc') -- 2
INSTR(s,s1)	从字符串s中获取s1的开始位置 SELECT INSTR('abc','b') -- 2
REVERSE(s)	将字符串s的顺序反过来 SELECT REVERSE('abc') -- cba
ELT(n,s1,s2,...)	返回第n个字符串 SELECT ELT(2,'a','b','c') -- b
EXPORT_SET(x,s1,s2)	返回一个字符串，在这里对于在“bits”中设定每一位，你得到一个“on”字符串，并且对于每个复位(reset)的位，你得到一个“off”字符串。每个字符串用“separator”分隔(缺省“,”)，并且只有“bits”的“number_of_bits” (缺省64)位被使用。 SELECT EXPORT_SET(5,'Y','N',' ',4) -- Y,N,Y,N
FIELD(s,s1,s2...)	返回第一个与字符串s匹配的字符串位置 SELECT FIELD('c','a','b','c') -- 3
FIND_IN_SET(s1,s2)	返回在字符串s2中与s1匹配的字符串的位置

MAKE_SET(x,s1,s2)	<p>返回一个集合 (包含由 “,” 字符分隔的子串组成的一个 字符串), 由相应的位在bits集合中的的字符串组成。str1对应于位0, str2对 应位1, 等等。</p> <p>SELECT MAKE_SET(1 4,'a','b','c'); -- a,c</p>
SUBSTRING_INDEX	<p>返回从字符串str的第count个出现的分隔符delim之后的子串。</p> <p>如果count是正数, 返回第count个字符左边的字符串。</p> <p>如果count是负数, 返回第(count的绝对值(从右边数))个字符右边的字符串。</p> <p>SELECT SUBSTRING_INDEX('a*b','*',1) -- a SELECT SUBSTRING_INDEX('a*b','*','-1) -- b SELECT SUBSTRING_INDEX(SUBSTRING_INDEX('a*b*c*d*e','*',3),'*','-1) -- c</p>
LOAD_FILE(file_name)	<p>读入文件并且作为一个字符串返回文件内容。文件必须在服务器上, 你必须指定到文件的完整路径名, 而且你必须有file权限。文件必须所有内容都是可读的并且小于max_allowed_packet。 如果文件不存在或由于上面原因之一不能被读出, 函数返回NULL。</p>

三、日期时间函数

MySQL的日期和时间函数主要用于处理日期时间。

函数	说明
CURDATE(),CURRENT_DATE()	<p>返回当前日期</p> <p>SELECT CURDATE() ->2014-12-17</p>
CURTIME(),CURRENT_TIME	<p>返回当前时间</p> <p>SELECT CURTIME() ->15:59:02</p>
NOW(),CURRENT_TIMESTAMP(),LOCALTIME(), SYSDATE(),LOCALTIMESTAMP()	<p>返回当前日期和时间</p> <p>SELECT NOW() ->2014-12-17 15:59:02</p>
UNIX_TIMESTAMP()	<p>以UNIX时间戳的形式返回当前时间</p> <p>SELECT UNIX_TIMESTAMP() ->1418803177</p>
	<p>将时间d以UNIX时间戳的形式返回</p>

UNIX_TIMESTAMP(d)	SELECT UNIX_TIMESTAMP('2011-11-11 11:11:11') ->1320981071
FROM_UNIXTIME(d)	将UNIX时间戳的时间转换为普通格式的时间 SELECT FROM_UNIXTIME(1320981071) ->2011-11-11 11:11:11
UTC_DATE()	返回UTC日期 SELECT UTC_DATE() ->2014-12-17
UTC_TIME()	返回UTC时间 SELECT UTC_TIME() ->08:01:45 (慢了8小时)
MONTH(d)	返回日期d中的月份值，1->12 SELECT MONTH('2011-11-11 11:11:11') ->11
MONTHNAME(d)	返回日期当中的月份名称，如January SELECT MONTHNAME('2011-11-11 11:11:11') ->November
DAYNAME(d)	返回日期d是星期几，如Monday,Tuesday SELECT DAYNAME('2011-11-11 11:11:11') ->Friday
DAYOFWEEK(d)	日期d今天是星期几，1星期日，2星期一 SELECT DAYOFWEEK('2011-11-11 11:11:11') ->6
WEEKDAY(d)	日期d今天是星期几， 0表示星期一，1表示星期二
WEEK(d), WEEKOFYEAR(d)	计算日期d是本年的第几个星期，范围是0->53 SELECT WEEK('2011-11-11 11:11:11') ->45
DAYOFYEAR(d)	计算日期d是本年的第几天 SELECT DAYOFYEAR('2011-11-11 11:11:11') ->315

DAYOFMONTH(d)	<p>计算日期d是本月的第几天</p> <p>SELECT DAYOFMONTH('2011-11-11 11:11:11')</p> <p>->11</p>
QUARTER(d)	<p>返回日期d是第几季节，返回1->4</p> <p>SELECT QUARTER('2011-11-11 11:11:11')</p> <p>->4</p>
HOUR(t)	<p>返回t中的小时值</p> <p>SELECT HOUR('1:2:3')</p> <p>->1</p>
MINUTE(t)	<p>返回t中的分钟值</p> <p>SELECT MINUTE('1:2:3')</p> <p>->2</p>
SECOND(t)	<p>返回t中的秒钟值</p> <p>SELECT SECOND('1:2:3')</p> <p>->3</p>
EXTRACT(type FROM d)	<p>从日期d中获取指定的值，type指定返回的值</p> <p>SELECT EXTRACT(MINUTE FROM '2011-11-11 11:11:11')</p> <p>->11</p> <p>type可取值为:</p> <p>MICROSECOND</p> <p>SECOND</p> <p>MINUTE</p> <p>HOUR</p> <p>DAY</p> <p>WEEK</p> <p>MONTH</p> <p>QUARTER</p> <p>YEAR</p> <p>SECOND_MICROSECOND</p> <p>MINUTE_MICROSECOND</p> <p>MINUTE_SECOND</p> <p>HOUR_MICROSECOND</p> <p>HOUR_SECOND</p> <p>HOUR_MINUTE</p> <p>DAY_MICROSECOND</p> <p>DAY_SECOND</p> <p>DAY_MINUTE</p>

	DAY_HOUR YEAR_MONTH
TIME_TO_SEC(t)	将时间t转换为秒 SELECT TIME_TO_SEC('1:12:00') ->4320
SEC_TO_TIME(s)	将以秒为单位的时间s转换为时分秒的格式 SELECT SEC_TO_TIME(4320) ->01:12:00
TO_DAYS(d)	计算日期d距离0000年1月1日的天数 SELECT TO_DAYS('0001-01-01 01:01:01') ->366
FROM_DAYS(n)	计算从0000年1月1日开始n天后的日期 SELECT FROM_DAYS(1111) ->0003-01-16
DATEDIFF(d1,d2)	计算日期d1->d2之间相隔的天数 SELECT DATEDIFF('2001-01-01','2001-02-02') ->-32
ADDDATE(d,n)	计算其实日期d加上n天的日期
ADDDATE(d, INTERVAL expr type)	计算起始日期d加上一个时间段后的日期 SELECT ADDDATE('2011-11-11 11:11:11',1) ->2011-11-12 11:11:11 (默认是天) SELECT ADDDATE('2011-11-11 11:11:11', INTERVAL 5 MINUTE) ->2011-11-11 11:16:11 (TYPE的取值与上面那个列出来的函数类似)
DATE_ADD(d,INTERVAL expr type)	同上
SUBDATE(d,n)	日期d减去n天后的日期 SELECT SUBDATE('2011-11-11 11:11:11', 1) ->2011-11-10 11:11:11 (默认是天)
SUBDATE(d,INTERVAL expr type)	日期d减去一个时间段后的日期 SELECT SUBDATE('2011-11-11 11:11:11', INTERVAL 5 MINUTE)

	->2011-11-11 11:06:11 (TYPE的取值与上面那个列出来的函数类似)
ADDTIME(t,n)	时间t加上n秒的时间 SELECT ADDTIME('2011-11-11 11:11:11', 5) ->2011-11-11 11:11:16 (秒)
SUBTIME(t,n)	时间t减去n秒的时间 SELECT SUBTIME('2011-11-11 11:11:11', 5) ->2011-11-11 11:11:06 (秒)
DATE_FORMAT(d,f)	按表达式f的要求显示日期d SELECT DATE_FORMAT('2011-11-11 11:11:11','%Y-%m-%d %r') ->2011-11-11 11:11:11 AM
TIME_FORMAT(t,f)	按表达式f的要求显示时间t SELECT TIME_FORMAT('11:11:11','%r') 11:11:11 AM
GET_FORMAT(type,s)	获得国家地区时间格式函数 select get_format(date,'usa') ->%m.%d.%Y (注意返回的就是这个奇怪的字符串(format字符串))

四、条件判断函数

1、IF(expr,v1,v2)函数

如果表达式expr成立，返回结果v1；否则，返回结果v2。

```
SELECT IF(1 > 0, '正确', '错误')
->正确
```

2、IFNULL(v1,v2)函数

如果v1的值不为NULL，则返回v1，否则返回v2。

```
SELECT IFNULL(null, 'Hello Word')
->Hello Word
```

3、CASE

语法1:



CASE

```
WHEN e1
THEN v1
WHEN e2
THEN e2
...
ELSE vn
```

END



CASE表示函数开始，END表示函数结束。如果e1成立，则返回v1,如果e2成立，则返回v2，当全部不成立则返回vn，而当有一个成立之后，后面的就不执行了。



SELECT CASE

```
WHEN 1 > 0
THEN '1 > 0'
WHEN 2 > 0
THEN '2 > 0'
ELSE '3 > 0'
END
```

->1 > 0



语法2:

CASE expr

```
WHEN e1 THEN v1
WHEN e1 THEN v1
...
ELSE vn
```

END

如果表达式expr的值等于e1，返回v1；如果等于e2,则返回e2。否则返回vn。

SELECT CASE 1

```
WHEN 1 THEN '我是1'
WHEN 2 THEN '我是2'
ELSE '你是谁'
```

五、系统信息函数

系统信息函数用来查询MySQL数据库的系统信息。

函数	作用
	返回数据库的版本号

VERSION()	SELECT VERSION() ->5.0.67-community-nt
CONNECTION_ID()	返回服务器的连接数
DATABASE()、SCHEMA	返回当前数据库名
USER()、SYSTEM_USER()、SESSION_USER()、CURRENT_USER()、CURRENT_USER	返回当前用户
CHARSET(str)	返回字符串str的字符集
COLLATION(str)	返回字符串str的字符排列方式
LAST_INSERT_ID()	返回最近生成的AUTO_INCREMENT值

六、加密函数

加密函数是MySQL用来对数据进行加密的函数。

1、PASSWORD(str)

该函数可以对字符串str进行加密，一般情况下，PASSWORD(str)用于给用户的密码加密。

```
SELECT PASSWORD('123')
-->*23AE809DDACAF96AF0FD78ED04B6A265E05AA257
```

2、MD5

MD5(str)函数可以对字符串str进行散列，可以用于一些普通的不需要解密的数据加密。

```
SELECT md5('123')
-->202cb962ac59075b964b07152d234b70
```

3、ENCODE(str,pswd_str)与DECODE(crypt_str,pswd_str)

ENCODE函数可以使用加密密码pswd_str来加密字符串str，加密结果是二进制数，需要使用BLOB类型的字段保存。该函数与DECODE是一对，需要同样的密码才能够解密。

```
SELECT ENCODE('123','xxoo')
-->vx
SELECT DECODE(';vx','xxoo')
-->123
```

七、其他函数

1、格式化函数FORMAT(x,n)

FORMAT(x,n)函数可以将数字x进行格式化，将x保留到小数点后n位。

```
SELECT FORMAT(3.1415926,3)
-->3.142
```

2、不同进制的数字进行转换

- ASCII(s) 返回字符串s的第一个字符的ASCII码;
- BIN (x) 返回x的二进制编码;
- HEX(x) 返回x的十六进制编码;
- OCT(x) 返回x的八进制编码;
- CONV(x,f1,f2) 返回f1进制数变成f2进制数;

3、IP地址与数字相互转换的函数

- INET_ATON(IP)函数可以将IP地址转换为数字表示; IP值需要加上引号;
- INET_NTOA(n)函数可以将数字n转换成IP形式。

```
SELECT INET_ATON('192.168.0.1')
      ->3232235521
SELECT INET_NTOA(3232235521)
      ->192.168.0.1
```

4、加锁函数和解锁函数

- GET_LOCK(name,time)函数定义一个名称为nam、持续时间长度为time秒的锁。如果锁定成功, 则返回1; 如果尝试超时, 则返回0; 如果遇到错误, 返回NULL。
- RELEASE_LOCK(name)函数解除名称为name的锁。如果解锁成功, 则返回1; 如果尝试超时, 返回0了如果解锁失败, 返回NULL;
- IS_FREE_LOCK(name)函数判断是否已使用名为name的锁定。如果使用, 返回0, 否则, 返回1;

```
SELECT GET_LOCK('MySQL',10)
      ->1      (持续10秒)
SELECT IS_FREE_LOCK('MySQL')
      ->1
SELECT RELEASE_LOCK('MySQL')
      ->1
```

5、重复执行指定操作的函数

BENCHMARK(count.expr)函数将表达式expr重复执行count此, 然后返回执行时间。该函数可以用来判断MySQL处理表达式的速度。

```
SELECT BENCHMARK(10000,NOW())
      ->0      返回系统时间1万
```

6、改变字符集的函数

CONVERT(s USING cs)函数将字符串s的字符集变成cs。

```
SELECT CHARSET('ABC')
      ->utf-8

SELECT CHARSET(CONVERT('ABC' USING gbk))
      ->gbk
```

7、转换数据类型

- CAST(x AS type)
- CONVERT(x,type)

这两个函数只对BINARY、CHAR、DATE、DATETIME、TIME、SIGNED INTEGER、UNSIGNED INTEGER。



```
SELECT CAST('123' AS UNSIGNED INTEGER) + 1  
-->124
```

```
SELECT '123' + 1  
-->124 其实MySQL能默认转换
```

```
SELECT CAST(NOW() AS DATE)  
-->2014-12-1
```

来源: <http://www.cnblogs.com/kissdodog/p/4168721.html>