
8.1.2 系统函数的使用

MySQL 函数会对传递进来的参数进行处理，并返回一个处理结果，也就是返回一个值。MySQL 常用函数大概包括数值型函数、字符串型函数、日期时间函数、聚合函数等。

1. 数值型函数

(1) ABS 求绝对值；

例如：SELECT ABS(5), ABS(-0.4), ABS(-14), ABS(0)；

(2) SQRT 求二次方根；

例如：SELECT SQRT(36), SQRT(110), SQRT(-4)；

注：负数没有平方根，返回的结果为 NULL。

(3) MOD 求余数；

例如：SELECT MOD(69, 8), MOD(120, 6), MOD(15.8, 3)；

(4) CEIL 和 CEILING

两个函数功能相同，都是返回不小于参数的最小整数，即向上取整；

例如：SELECT CEIL(-4.5), CEILING(4.5)；

返回结果为-4 和 5。

(5) FLOOR 向下取整，返回值转化为一个 BIGINT；

例如：SELECT FLOOR(8), FLOOR(8.66), FLOOR(-8), FLOOR(-8.66)；

(6) RAND 生成一个 0~1 之间的随机数，传入整数参数是，用来产生重复序列；

例如：SELECT RAND(), RAND(), RAND()；

(7) ROUND 对所传参数进行四舍五入；

ROUND(x) 函数返回最接近于参数 x 的整数；ROUND(x, y) 函数对参数 x 进行四舍五入的操作，返回值保留小数点后面指定的 y 位

例如：SELECT ROUND(-3.33, 1), ROUND(3.33, 3), ROUND(33.33, -1), ROUND(33.33, -2)；

返回结果为-3.3、3.330、30 和 0。

(8) SIGN 返回参数的符号；

例如：SELECT SIGN(-9), SIGN(0), SIGN(33)；

(9) POW 和 POWER 两个函数的功能相同，都是所传参数的次方的结果值，如果是 POW(x, y) 函数和 POWER(x, y) 函数，即用于计算 x 的 y 次方。

例如：SELECT POW(10, -2), POW(10, 3), POW(10, 0), POWER(2, 3), POWER(2, -3)；

返回结果为 0.01、1000、1、8 和 0.125。

2. 字符串函数

(1) LENGTH 计算字符串长度函数，返回字符串的字节长度；

使用 utf8 (UNICODE 的一种变长字符编码) 编码字符集时，一个汉字是 3 个字节，一个数字或字母是一个字节；

例如：SELECT LENGTH('database'), LENGTH('数据库原理')；

(2) CONCAT 合并字符串函数，返回结果为连接参数产生的字符串；

若有任何一个参数为 NULL，则返回值为 NULL。若所有参数均为非二进制字符串，则结果为非二进制字符串。若自变量中含有任一二进制字符串，则结果为一个二进制字符串。

例如：SELECT CONCAT('data', 'base'), CONCAT('data', NULL)；

返回结果为 database 和 NULL。

(3) INSERT 替换字符串函数；

INSERT(s1, x, len, s2) 返回字符串 s1, 子字符串起始于 x 位置, 并且用 len 个字符长的字符串代替 s2。若 x 超过字符串长度, 则返回值为原始字符串。假如 len 的长度大于其他字符串的长度, 则从位置 x 开始替换。若任何一个参数为 NULL, 则返回值为 NULL。

例如: SELECT INSERT('database', 2, 4, 'LIB') AS a1, INSERT('database', -1, 4, 'LIB') AS a2, INSERT('database', 3, 20, 'LIB') AS a3, INSERT('database', NULL, 4, 'LIB') AS a4;
返回结果为 dLIBase、database、daLIB 和 NULL。

(4) LOWER 将字符串中的字母转换为小写;

例如: SELECT LOWER('DATABASE'), LOWER('database');

(5) UPPER 将字符串中的字母转换为大写;

例如: SELECT UPPER('DATABASE'), UPPER('database');

(6) LEFT 从左侧截取字符串, 返回字符串左边的若干个字符;

LEFT(s, n) 函数返回字符串 s 最左边的 n 个字符。

例如: SELECT LEFT('database', 2);

(7) RIGHT 从右侧截取字符串, 返回字符串右边的若干个字符;

RIGHT(s, n) 函数返回字符串 s 最右边的 n 个字符。

例如: SELECT RIGHT('database', 2);

(8) TRIM 删除字符串左右两侧的空格;

例如: SELECT ' [database] ', CONCAT(' [', TRIM(' database '), '] ');

(9) REPLACE 字符串替换函数, 返回替换后的新字符串;

REPLACE(s, s1, s2) 使用字符串 s2 替换字符串 s 中所有的字符串 s1。

例如: SELECT REPLACE('aaa.mysql.com', 'a', 'w');

(10) SUBSTRING 截取字符串, 返回从指定位置开始的指定长度的字符串;

SUBSTRING(s, n, len) 表示从字符串 s 返回一个长度为 len 字符相同的子字符串, 起始于位置 n。如果 n 是一个负值, 则子字符串的位置起始于字符串结尾的第 n 个字符, 即倒数第 n 个字符, 而不是字符串的开头位置。

例如: SELECT SUBSTRING('database', 3) AS col1, SUBSTRING('database', 3, 4) AS col2, SUBSTRING('database', -3) AS col3, SUBSTRING('database', -5, 3) AS col4;

返回结果为 tabase、taba、ase 和 aba。

(11) REVERSE 字符串反转 (逆序) 函数, 返回与原始字符串顺序相反的字符串;

例如: SELECT REVERSE('database');

3. 日期和时间函数

(1) CURDATE 和 CURRENT_DATE 两个函数作用相同, 返回当前系统的日期值;

例如: SELECT CURDATE(), CURRENT_DATE(), CURRENT_DATE()+0, CURRENT_DATE()+10;

注: CURDATE()+0 相当于将当前日期值转换为数值型。

(2) CURTIME 和 CURRENT_TIME 两个函数作用相同, 返回当前系统的时间值;

例如: SELECT CURTIME(), CURRENT_TIME(), CURRENT_TIME()+0, CURRENT_TIME()+10;

注: CURRENT_TIME()+0 相当于将当前时间转换为数值型。

(3) NOW 和 SYSDATE 两个函数作用相同, 返回当前系统的日期和时间值;

例如: SELECT NOW(), SYSDATE(), SYSDATE()+0, SYSDATE()+0;

注: SYSDATE()+0 相当于将当前日期时间转换为数值型。

(4) MONTH 获取指定日期中的月份;

例如: SELECT MONTH('2020-12-15');

(5) MONTHNAME 获取指定日期中的月份英文名称;

例如: SELECT MONTHNAME('2020-12-15');

(6) DAYNAME 获取指定日期对应的星期几的英文名称;

例如: SELECT DAYNAME('2020-12-15');

(7) DAYOFWEEK 获取指定日期对应的一周的索引位置值;

DAYOFWEEK 的索引: 1 表示周日, 2 表示周一, …… , 7 表示周六。

WEEKDAY 获取指定日期在一周内的对应的工作日索引;

WEEKDAY 的工作日索引: 0 表示周一, 1 表示周二, …… , 6 表示周日。

例如: SELECT DAYOFWEEK('2020-12-15'), WEEKDAY('2020-12-15');

(8) WEEK 获取指定日期是一年中的第几周, 返回值的范围是否为 0~52 或 1~53;

例如: SELECT WEEK('2020-12-15');

(9) DAYOFYEAR 获取指定日期是一年中的第几天, 返回值范围是 1~366;

例如: SELECT DAYOFYEAR('2020-12-15');

(10) DAYOFMONTH 获取指定日期是一个月中是第几天, 返回值范围是 1~31;

例如: SELECT DAYOFMONTH('2020-12-15');

(11) YEAR 获取年份, 返回值范围是 1970~2069;

例如: SELECT YEAR('2020-12-15');

(12) TIME_TO_SEC 将时间参数转换为秒数;

TIME_TO_SEC(time) 函数返回将参数 time 转换为秒数的时间值, 转换公式为 “小时×3600+分钟×60+秒”。

例如: SELECT TIME_TO_SEC('02:02:02');

(13) SEC_TO_TIME 将秒数转换为时间, 与 TIME_TO_SEC 互为反函数;

例如: SELECT SEC_TO_TIME('7322');

(14) DATEDIFF 获取两个日期之间间隔;

DATEDIFF(date1, date2) 返回起始时间 date1 和结束时间 date2 之间的天数。date1 和 date2 为日期或 date-and-time 表达式。计算时只用到这些值的日期部分。

例如: SELECT DATEDIFF('2020-11-30', '2020-10-30') AS a1, DATEDIFF('2020-11-30', '2020-12-30') AS a2;

4. 聚合函数

(1) MAX 查询指定列的最大值

(2) MIN 查询指定列的最小值

(3) COUNT 统计查询结果的行数

(4) SUM 求和, 返回指定列的总和

(5) AVG 求平均值, 返回指定列数据的平均值

5. 流程控制函数

(1) IF 判断, 流程控制, 语法结构如下:

IF(expr, v1, v2)

其中: 表达式 expr 得到不同的结果, 当 expr 为真是返回 v1 的值, 否则返回 v2。

例如: SELECT IF(1<2, 1, 0) a1, IF(STRCMP('abc', 'ab'), 'yes', 'no') a2;

【例 8-4】显示 reader 表中读者的姓名和性别, 性别为 '女' 显示为 0, 否则显示为 1, 只显示前 10 行, 在命令行窗口中输入如下命令:

```
select readerName 姓名, if(gender='女', 0, 1) 性别 from reader limit 10;
```

```
mysql> select readerName 姓名,if(gender='女',0,1) 性别 from reader limit 10;
```

姓名	性别
李瑞	1
陈梓曦	0
崔同贺	1

(2) IFNULL 判断是否为空;

IFNULL 函数接受两个参数,如果不是 NULL,则返回第一个参数。否则,IFNULL 函数返回第二个参数。两个参数可以是文字值或表达式。函数的语法如下:

```
IFNULL(v1,v2);
```

其中:如果 v1 不为 NULL,则 IFNULL 函数返回 v1;否则返回 v2 的结果。

【例 8-5】查看 borrow 表中读者编号为'r003'的还书情况,已有还书日期的显示还书日期,否则显示为'未还书'。在命令行窗口中输入如下命令:

```
select readerNo 读者编号,bookNo 书号,IFNULL(returnDate,'未还书') 还书情况
from borrow where readerNo='r003';
```

```
mysql> select readerNo 读者编号,bookNo 书号,IFNULL(returnDate,'未还书') 还书情况 from borrow where readerNo='r003';
```

读者编号	书号	还书情况
r003	b109	2017-07-31 00:00:00
r003	b080	2017-10-16 00:00:00
r003	b143	2019-05-27 00:00:00
r003	b127	2019-09-05 00:00:00
r003	b110	2019-10-31 00:00:00
r003	b144	2018-06-11 00:00:00
r003	b010	2018-10-27 00:00:00
r003	b003	2021-01-10 00:00:00
r003	b003	2021-01-10 00:00:00
r003	b003	2021-01-10 00:00:00
r003	b003	未还书
r003	b120	未还书

12 rows in set (0.05 sec)

(3) CASE 搜索判断语句,语法如下:

```
CASE
  WHEN <条件 1> THEN <命令>
  WHEN <条件 2> THEN <命令>
  ...
  ELSE commands
END CASE;
```

MySQL 分别计算 WHEN 子句中的每个条件,直到找到一个值为 TRUE 的条件,然后执行 THEN 子句中的相应<命令>。如果没有一个条件为 TRUE,则执行 ELSE 子句中的<命令>。如果不指定 ELSE 子句,并且没有一个条件为 TRUE,MySQL 将发出错误消息。MySQL 不允许在 THEN 或 ELSE 子句中使用空的命令。

【例 8-6】查看 book 表中图书的在架情况,如果该书没有在架,则显示为'零本',只剩 1 本在架,则显示为'孤本',少于 6 本则显示为'少量',否则显示为'大量',只显示前 10 本书的情况。在命令行窗口中输入如下命令:

```
select bookNo 书号,bookname 书名,
case
  when number=0 then '零本'
  when number=1 then '孤本'
  when number>1 and number<6 then '少量'
  else '大量'
end as 数量
```

from book limit 10;

```
mysql> select bookNo 书号,bookname 书名,  
-> case  
->   when number=0 then '零本'  
->   when number=1 then '孤本'  
->   when number>1 and number<6 then '少量'  
->   else '大量'  
-> end as 数量  
-> from book limit 10;
```

书号	书名	数量
b001	数据库系统原理与设计	少量
b002	JAVA	大量
b003	红楼梦	大量
b004	英语写作	大量
b005	最漫画	少量
b006	论中国农业剩余劳动力转移-农业现代化的必由之路	孤本
b007	明清时期山东商品经济的发展	少量
b008	发展经济学	大量
b009	粮食经济中的和谐-中国粮食市场与政府宏观政策的耦合	孤本
b010	经济体制的比较制度分析	少量

10 rows in set (0.00 sec)