

# 数据库函数功能与用法

---

## 1.字符串函数

---

### 1.1 ASCII()

#### 1.1.1 功能

返回字符串第一个字符的ASCII码

#### 1.1.2 使用

```
mysql> SELECT user,ASCII(user) AS asione FROM t1;
+-----+-----+
| user   | asione |
+-----+-----+
| xiaobai |      120 |
+-----+-----+
1 row in set (0.00 sec)
```

### 1.2 CHAR\_LENGTH()和 CHARACTER\_LENGTH()

#### 1.2.1 功能

返回字符串字符数

#### 1.2.2 使用

```
mysql> SELECT user,CHAR_LENGTH(user) AS numuser FROM t1;
+-----+-----+
| user   | numuser |
+-----+-----+
| xiaobai |        7 |
+-----+-----+
```

```
mysql> SELECT user,CHARACTER_LENGTH(user) AS numuser FROM t1;
+-----+-----+
| user   | numuser |
+-----+-----+
| xiaobai |        7 |
+-----+-----+
```

### 1.3 CONCAT()

#### 1.3.1 功能

合并多个字符串为一个字符串

#### 1.3.2 使用

```
mysql> SELECT CONCAT("Hello","Wolrd","!") AS wholestring;
+-----+
| wholestring |
+-----+
| HelloWorld! |
+-----+
```

```
mysql> SELECT id,name,pwd,CONCAT(t3.id,t3.name,t3.pwd) AS wholestring FROM t3;
+-----+-----+-----+-----+
| id | name      | pwd                                | wholestring |
+-----+-----+-----+-----+
| 1 | xiaoxiang | 9766527f2b5d3e95d4a733fcfb77bd7e | 1xiaoxiang9766527f2b5d3e95d4a733fcfb77bd7e |
| 2 | xiaonan  | ab233b682ec355648e7891e66c54191b | 2xiaonanab233b682ec355648e7891e66c54191b |
| 3 | xiaowei  | 22ac3c5a5bf0b520d281c122d1490650 | 3xiaowei22ac3c5a5bf0b520d281c122d1490650 |
+-----+-----+-----+-----+
```

## 1.4 CONCAT\_WS()

### 1.4.1 功能

类似CONCAT()函数，但可给每个字符串之间增加分隔符

### 1.4.2 使用

```
mysql> SELECT CONCAT_WS("?", "Hello", "Wolrd", "!") AS wholestring;
+-----+
| wholestring |
+-----+
| Hello?Wolrd?! |
+-----+
```

## 1.5 FIELD()

### 1.5.1 功能

#### 1.5.1.1

返回第一个字符串在字符串列表中的位置

#### 1.5.1.2

与ORDER BY混用，进行自定义排序

### 1.5.2 使用

#### 1.5.2.1

```
mysql> SELECT FIELD("C","A","B","C","D","F","E");
+-----+
| FIELD("C","A","B","C","D","F","E") |
+-----+
| 3 |
+-----+
```

#### 1.5.2.2

```
mysql> SELECT * FROM t4 ORDER BY FIELD(name,'xiaohua','xiaobai') DESC;
```

id	name	pwd	age
3	xiaobai	444	21
5	xiaobai	456	21
1	xiaohua	123	12
4	xiaohua	123456	16
2	xiaolong	489	19
6	xiaohu	856	21

## 1.6 FIND\_IN\_SET(s1,s2)

### 1.6.1 功能

返回在字符串s2中与字符串s1匹配的字符串的位置

### 1.6.2 使用

```
mysql> SELECT FIND_IN_SET("C","A,B,C,D,E,F");
```

FIND_IN_SET("C","A,B,C,D,E,F")
3

## 1.7 FORMAT(x,n)

### 1.7.1 功能

可以将数字x格式化为“#,###.###”的格式，且x保留小数点后n位，最后一位四舍五入

### 1.7.2 使用

```
mysql> SELECT FORMAT(25635.3325,2);
```

FORMAT(25635.3325,2)
25,635.33

## 1.8 INSERT(s1,x,len,s2)

### 1.8.1 功能

字符串s2替换字符串s1中从位置x开始之后的len个字符

### 1.8.2 使用

```
mysql> SELECT INSERT("www.google.com",5,6,"apples");
+-----+
| INSERT("www.google.com",5,6,"apples") |
+-----+
| www.apples.com                        |
+-----+
```

## 1.9 LOCATE(s1,s2)和POSITION(s1 IN s2)

### 1.9.1 功能

从s2中获取s1在s2中的开始位置

### 1.9.2 使用

```
mysql> SELECT LOCATE("pl","apples");
+-----+
| LOCATE("pl","apples") |
+-----+
| 3 |
+-----+
```

```
mysql> SELECT POSITION("pl" IN "apples");
+-----+
| POSITION("pl" IN "apples") |
+-----+
| 3 |
+-----+
```

## 1.10 LCASE()和LOWER()

### 1.10.1 功能

将字符串的字母都转换成小写字母

### 1.10.2 使用

```
mysql> SELECT LCASE("APPLES1");
+-----+
| LCASE("APPLES1") |
+-----+
| apples1          |
+-----+
```

```
mysql> SELECT LOWER("APPLES1");
+-----+
| LOWER("APPLES1") |
+-----+
| apples1          |
+-----+
```

###

## 1.11 LEFT(s,n)

### 1.11.1 功能

返回字符串s前n个字符

### 1.11.2 使用

```
mysql> SELECT LEFT("apples",3);
+-----+
| LEFT("apples",3) |
+-----+
| app              |
+-----+
```

## 1.12 LPAD(s1,len,s2)和RPAD(s1,len,s2)

### 1.12.1 功能

L:在s1的开始处填充s2使得字符串长度达到len

R:在s1的结尾处填充s2使得字符串长度达到len

### 1.12.2 使用

```
mysql> SELECT LPAD("les",6,"app");
+-----+
| LPAD("les",6,"app") |
+-----+
| apples              |
+-----+

mysql> SELECT RPAD('app',6,'les');
+-----+
| RPAD('app',6,'les') |
+-----+
| apples              |
+-----+
```

## 1.13 LTRIM()和RTRIM()和TRIM()

### 1.13.1 功能

L:去掉字符串开始的空格

R:去掉字符串结尾的空格

T:去掉开始、结尾的空格

### 1.13.2 使用

```
mysql> SELECT LTRIM("      apples");
+-----+
| LTRIM("      apples") |
+-----+
| apples                |
+-----+
```



```
mysql> SELECT RTRIM('apples ');
+-----+
| RTRIM('apples ') |
+-----+
| apples           |
+-----+
```

```
mysql> SELECT TRIM('  apples ');
+-----+
| TRIM('  apples ') |
+-----+
| apples           |
+-----+
```

## 1.14 MID/SUBSTRING(s,n,len)

### 1.14.1 功能

从字符串s的n位置开始截取长度为len的子字符串

### 1.14.2 使用

```
mysql> SELECT MID("badapples",4,6);
+-----+
| MID("badapples",4,6) |
+-----+
| apples              |
+-----+
```

## 1.15 REPEAT(s,n)

### 1.15.1 功能

将字符串s重复n遍

### 1.15.2 使用

```
mysql> SELECT REPEAT("apples",3);
+-----+
| REPEAT("apples",3) |
+-----+
| applesapplesapples |
+-----+
```

## 1.16 REPLACE(s,s1,s2)

### 1.16.1 功能

用s2替换s中的s1

### 1.16.2 使用

```
mysql> SELECT REPLACE('abc','a','x');
+-----+
| REPLACE('abc','a','x') |
+-----+
| xbc          |
+-----+
```

## 1.17 REVERSE()

### 1.17.1 功能

将字符串倒序

### 1.17.2 使用

```
mysql> SELECT REVERSE('abc');
+-----+
| REVERSE('abc') |
+-----+
| cba          |
+-----+
```

## 1.18 RIGHT(s,n)

### 1.18.1 功能

返回s后n个字符

### 1.18.2 使用

```
mysql> SELECT RIGHT('badapples',3);
+-----+
| RIGHT('badapples',3) |
+-----+
| les          |
+-----+
```

## 1.19 SPACE()

### 1.19.1 功能

返回n个空格

### 1.19.2 使用

```
mysql> SELECT RPAD(SPACE(5),10,'apple');
+-----+
| RPAD(SPACE(5),10,'apple') |
+-----+
|      apple          |
+-----+
```

## 1.20 STRCMP(s1,s2)

### 1.20.1 功能

比较s1和s2:

如果s1=s2, 返回0

s1>s2, 返回1

s1<s2, 返回-1

### 1.20.2 使用

```
mysql> SELECT STRCMP("apples","badapples");
+-----+
| STRCMP("apples","badapples") |
+-----+
| -1 |
+-----+
mysql> SELECT STRCMP("apples","alples");
+-----+
| STRCMP("apples","alples") |
+-----+
| 1 |
+-----+
mysql> SELECT STRCMP("apples","badapples");
+-----+
| STRCMP("apples","badapples") |
+-----+
| -1 |
+-----+
mysql> SELECT STRCMP("apples","alples");
+-----+
| STRCMP("apples","alples") |
+-----+
| 1 |
+-----+
```

## 1.21 SUBSTRING()或SUBSTR(s,start,length)

### 1.21.1 功能

从s的start位置截取长度为length的字符串

### 1.21.2 使用

```
mysql> SELECT SUBSTR("badapples",4,6);
+-----+
| SUBSTR("badapples",4,6) |
+-----+
| apples |
+-----+
```

## 1.22 SUBSTRING\_INDEX(s,delimiter,num)

### 1.22.1 功能



返回从s的第num个出现的分隔符delimiter之后的子串

如果num为正数，返回第num个分隔符左边的字符串

如果num为负数，返回第num（绝对值）个分隔符右边的字符串

### 1.22.2 使用

```
mysql> SELECT SUBSTRING_INDEX('a*b*c*d*e','*',4);
+-----+
| SUBSTRING_INDEX('a*b*c*d*e','*',4) |
+-----+
| a*b*c*d                             |
+-----+
```

## 1.23 UCASE()和UPPER()

### 1.23.1 功能

转换为大写

### 1.23.2 使用

```
mysql> SELECT UCASE('apples');
+-----+
| UCASE('apples') |
+-----+
| APPLES          |
+-----+
```

## 2. 数字函数

---

### 2.1 ABS(x)

#### 2.1.1 功能

返回x的绝对值

#### 2.1.2 使用

```
mysql> SELECT ABS(-8);
+-----+
| ABS(-8) |
+-----+
|      8  |
+-----+
```

### 2.2 ACOS(),ASIN(),ATAN(),ATAN2() COS(),COT(),TAN()

#### 2.2.1 功能

就是他们在数学中的含义，ATAN2()有点特殊,ATAN2(y,x)=TAN(y/x)

#### 2.2.2 使用

```
mysql> SELECT ATAN2(-7,7);
+-----+
| ATAN2(-7,7) |
+-----+
| -0.7853981633974483 |
+-----+
```

## 2.3 AVG()

### 2.3.1 功能

取平均值，括号里为一个字段

### 2.3.2 使用

```
mysql> select * from t2;
+----+-----+-----+
| id | user  | password |
+----+-----+-----+
| 1  | xihua | 15648465 |
| 4  | xihuo | 15645865 |
| 5  | NULL  | 4122cb13c7a474c1976c9706ae36521d |
+----+-----+-----+
3 rows in set (0.00 sec)

mysql> SELECT AVG(id) FROM t2;
+-----+
| AVG(id) |
+-----+
| 3.3333 |
+-----+
```

## 2.4 CEIL(x)/CEILING(x)和FLOOR(x)

### 2.4.1 功能

C:返回大于或等于x的最小整数

F:返回小于或等于x的最大整数

### 2.4.2 使用

```
mysql> SELECT CEIL(2.44);
+-----+
| CEIL(2.44) |
+-----+
| 3 |
+-----+
```

## 2.5 COUNT(expre)

### 2.5.1 功能

返回查询记录总数

### 2.5.2 使用

```
mysql> SELECT COUNT(id) FROM t2;
+-----+
| COUNT(id) |
+-----+
|          3 |
+-----+
```

## 2.6 DEGREES()

### 2.6.1 功能

将弧度转化为角度

### 2.6.2 使用

```
mysql> SELECT DEGREES(3.1415926);
+-----+
| DEGREES(3.1415926) |
+-----+
| 179.99999692953102 |
+-----+
```

## 2.7 n DIV m

### 2.7.1 功能

整除，n整除m

### 2.7.2 使用

```
mysql> SELECT 4.0 DIV 3.0;
+-----+
| 4.0 DIV 3.0 |
+-----+
|            1 |
+-----+
```

## 2.8 EXP(x)和LN(x)

### 2.8.1 功能

E:返回e的x次方

L:返回以e为底的自然对数

### 2.8.2 使用

```
mysql> SELECT EXP(2);
+-----+
| EXP(2) |
+-----+
| 7.38905609893065 |
+-----+
```

## 2.9 GREATEST(x1,x2,...,xn)和LEAST()

### 2.9.1 功能

G:返回列表中最大值

L:返回列表中最小值

### 2.9.2 使用

```
mysql> SELECT GREATEST(52,5,88,6,63,4,5,79,92);
+-----+
| GREATEST(52,5,88,6,63,4,5,79,92) |
+-----+
| 92 |
+-----+
```

## 2.10 LOG(base,x)

### 2.10.1 功能

返回自然对数，base为指定底数，如果没有，默认为e

## 2.11 LOG10(x)

### 2.11.1 功能

返回以10为底数的对数

## 2.12 MIN(expre)和MAX(expre)

### 2.12.1 功能

返回字段中的最小值或最大值

## 2.13 MOD(x,y)

### 2.13.1 功能

$x\%y$

## 2.14 PI()

### 2.14.1 功能

返回圆周率

## 2.15 POW(x,y)或POWER(x,y)

## 2.15.1 功能

返回x的y次方

## 2.16 RADIANS(x)

### 2.16.1 功能

将角度转化为弧度

## 2.17 RAND()

### 2.17.1 功能

返回0到1的随机数

### 2.17.2 使用

```
| RAND() |
+-----+
| 0.09694787775813404 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT RAND();
+-----+
| RAND() |
+-----+
| 0.1931847270514674 |
+-----+
```

## 2.18 ROUND(x)

### 2.18.1 功能

返回离x最近的整数

## 2.19 SIGN(x)

### 2.19.1 功能

返回x的符号，分别为-1,0,1

## 2.20 SQRT(x)

### 2.20.1 功能

返回x的平方根



## 2.21 SUM(expre)

### 2.21.1 功能

返回指定字段总和

## 2.22 TRUNCATE(x,y)

### 2.22.1 功能

返回数值x保留到小数点后y位，注意，这里是截断，不是四舍五入

### 2.22.2 使用

```
mysql> SELECT TRUNCATE(2.5986542,3);
+-----+
| TRUNCATE(2.5986542,3) |
+-----+
| 2.598 |
+-----+
```

## 3. 日期函数

---

### 3.1 ADDDATE(d,n)

#### 功能

起始日期d加n天

```
mysql> SELECT ADDDATE('2002-12-28',3);
+-----+
| ADDDATE('2002-12-28',3) |
+-----+
| 2002-12-31 |
+-----+
```

### 3.2 ADDTIME(t,n)

#### 功能

t加上时间表达式n

```
mysql> SELECT ADDTIME('2001-12-26 1:11:11','2:35:23');
+-----+
| ADDTIME('2001-12-26 1:11:11','2:35:23') |
+-----+
| 2001-12-26 03:46:34 |
+-----+
```

### 3.3 CURDATE()/CUETIME()

返回当前日期/时间

### 3.4 CURENT\_TIMESTAMP(), LOCALTIME(), LOCALTIMESTAMP(), NOW(), SYSDATE()

返回当前日期和时间

### 3.5 DATE(), TIME(), YEAR(), QUARTER(), MONTH(), WEEK(), DAY(), HOUR(), MINUTE(), SECOND(), MICROSECOND()

从日期或日期表达式中提取日期值

提取时间值

返回年份

返回季节值，从1到4

返回月份值，从1到12

返回日期为当年第几个星期，从0到53

返回日期值的日期部分

返回日期值的小时部分

返回日期值的分钟部分

返回日期值的秒数

提取日期值的微秒数

### 3.6 DATEDIFF(d1,d2), TIMEDIFF()

计算d1->d2相隔天数

计算时间差值

### 3.7 DATE\_FORMAT(d,f), TIME\_FORMAT(t,f)

#### 3.7.1 功能

按格式f显示日期d

按格式f显示时间t

#### 3.7.2 使用

```
mysql> SELECT DATE_FORMAT('2000-2-25 11:11:11', '%Y-%m-%d %r');
+-----+
| DATE_FORMAT('2000-2-25 11:11:11', '%Y-%m-%d %r') |
+-----+
| 2000-02-25 11:11:11 AM |
+-----+
```

### 3.8 DATE\_SUB(d, INTERVAL n type)

从日期d减去指定时间间隔n type

```
mysql> SELECT DATE_SUB('2002-2-25',INTERVAL 10 DAY);
+-----+
| DATE_SUB('2002-2-25',INTERVAL 10 DAY) |
+-----+
| 2002-02-15 |
+-----+
```

### 3.9 DAYNAME(d), MONTHNAME()

返回日期是星期几，如TUESDAY

返回日期是几月份，如NOVEMBER

### 3.10 DAYOFMONTH(), DAYOFYEAR(), WEEKOFYEAR()

返回日期是当月第几天

返回日期是当年第几天

返回日期是当年第几星期

### 3.11 DAYOFWEEK(), WEEKDAY()

返回日期当日为星期几，1是星期日，2是星期一，以此类推

返回日期当日为星期几，0是星期一，1是星期二，以此类推

### 3.12 EXTRACT(type FROM d)

从日期中提取指定值

p s:

type可取值为:

```
MICROSECOND SECOND MINUTE HOUR DAY WEEK MONTH QUARTER YEAR
SECOND_MICROSECOND MINUTE_MICROSECOND MINUTE_SECOND HOUR_MICROSECOND
HOUR_SECOND HOUR_MINUTE DAY_MICROSECOND DAY_SECOND DAY_MINUTE DAY_HOUR
YEAR_MONTH
```

### 3.13 FROM\_DAYS(n), TO\_DAYS()

返回从0000年1月1日开始n天后的日期

计算日期到0000年1月1日的天数

### 3.14 LAST\_DAY()

返回日期所在月最后一天的日期

### 3.15 MAKEDATE(year,day\_of\_year)

返回一日期，year为年份，day\_of\_year为天数序号

```
mysql> SELECT MAKEDATE(2018,285);
+-----+
| MAKEDATE(2018,285) |
+-----+
| 2018-10-12          |
+-----+
```

### 3.16 MAKETIME(h,m,s)

组合时间h, m, s为h: m: s

### 3.17 PERIOD\_ADD(period,n)

为 年-月 组合添加月份值

```
mysql> SELECT PERIOD_ADD(201806,6)
-> ;
+-----+
| PERIOD_ADD(201806,6) |
+-----+
| 201812                |
+-----+
```

### 3.18 SEC\_TO\_TIME(), TIME\_TO\_SEC()

将以秒为单位的时间以时分秒格式输出

将时间转换为秒数

### 3.19 STR\_TO\_DATE()

将字符串转变为日期

```
mysql> SELECT STR_TO_DATE('July 22 2016','%M %d %Y');
+-----+
| STR_TO_DATE('July 22 2016','%M %d %Y') |
+-----+
| 2016-07-22                              |
+-----+
```

### 3.20 SUBDATE(d,n), SUBTIME(t,n)

返回d减去n天后的日期

返回t减去n秒的时间

### 3.21 TIMESTAMP()

单个参数，返回相应日期或时间；

两个参数，组合后返回

```
mysql> SELECT TIMESTAMP('2018-5-16','16:25:29');
+-----+
| TIMESTAMP('2018-5-16','16:25:29') |
+-----+
| 2018-05-16 16:25:29 |
+-----+
```

## 3.22 YEARWEEK(date,mode)

返回年份和第几周（0-53）

mode中0表示周天，1表示周一，以此类推

```
mysql> SELECT YEARWEEK('2018-6-29');
+-----+
| YEARWEEK('2018-6-29') |
+-----+
| 201825 |
+-----+
```

```
mysql> SELECT YEARWEEK('2018-6-29',1);
+-----+
| YEARWEEK('2018-6-29',1) |
+-----+
| 201826 |
+-----+
```

```
mysql> SELECT YEARWEEK('2018-6-29',2);
+-----+
| YEARWEEK('2018-6-29',2) |
+-----+
| 201825 |
+-----+
```

ques:这里当mode=2返回值与真值相同，但是我查询了日历发现2018-6-29这天是星期五

p s：好吧，我懂了（ $5+2=7$ ,进一位）

p s：此函数可以用来猜日期

## 4. 高级函数

### 4.1 BIN(x)

返回x的二进制编码



```

+-----+
| BIN(6) |
+-----+
| 110    |
+-----+
1 row in set (0.01 sec)

mysql> SELECT BIN('apples');
+-----+
| BIN('apples') |
+-----+
| 0              |
+-----+

```

ps：此函数可判断字符串和具体数字

## 4.2 BINARY(s)

将s转换为二进制字符串（VARCHAR）

```

mysql> SELECT BINARY('apples');
+-----+
| BINARY('apples') |
+-----+
| apples            |
+-----+

```

## 4.3 CASE ep WHEN cd1 THEN result1 WHEN cd2 THEN result2 ... ELSE result END

CASE表示函数开始，END表示结束，当满足条件1（cd1）时，返回result1，以此类推，与switch语句相似

## 4.4 CAST(x,type)

转换数据类型

## 4.5 COALESCE(ep1,ep2,...,epn)

返回参数中第一个非空表达式（从左向右）

## 4.6 CONV(x,f1,f2)

返回f1进制数转换f2进制数

```

mysql> SELECT CONV(46,10,2);
+-----+
| CONV(46,10,2) |
+-----+
| 101110        |
+-----+

```

## 4.7 CONVERT(s USING cs)

cs为新编码方式

该函数改变字符集

## 4.8 CURRENT\_USER(), USER()

返回当前用户

## 4.9 DATABASE()

返回当前数据库名

## 4.10 IF(ep,v1,v2)

如果表达式ep成立，返回v1；否则，返回v2

## 4.11 IFNULL(v1,v2)

如果v1值不为NULL，返回v1；否则，返回v2

## 4.12 ISNULL()

判断表达式是否为NULL

## 4.13 LAST\_INSERT\_ID()

返回最近生成的AUTO\_INCREMENT值

```
mysql> SELECT LAST_INSERT_ID();
+-----+
| LAST_INSERT_ID() |
+-----+
| 0 |
+-----+
```

## 4.14 NULLIF(ep1,ep2)

比较两个字符串，如果ep1和ep2相等，返回NULL；否则返回ep1

## 4.15 VERSION()

返回数据库版本号