

## 数值类函数

---

- **ABS(X)**  
绝对值 ( `absolute` ) ;
- **CEIL(x), CEILING(x)**  
不小于x的最小整数;
- **FLOOR(x)**  
与 `CEIL(x)` , `CEILING(x)` 相反;
- **ROUND(x)**  
取整, 四舍五入;
- **RAND()**  
产生0-1的随机数;
- **RAND(x)**  
x为随机数种子, 次函数不管执行多少次, 只会产生一个固定的值;
- **SIGN(x)**  
求x的符号
- **PI()**  
圆周率
- **TRUNCATE(x,y)**  
保留 x 小数点后 y 位, 不四舍五入;
- **ROUND(x, y)**  
四舍五入到 y 位; **ROUND(x)** 其实就是 y 默认为 0 的情况;
- **POW(x,y), POWER(x,y)**  
幂计算;
- **SQRT(x)**  
平方根;
- **\*\* EXP(x)\*\***  
e的幂;
- **MOD(x,y)**  
余数计算;
- **LOG(x)**  
求自然 (e) 对数; 相似的还有**LOG10(x)** ;
- **RADIANS-弧度; DEGREES-角度;**
- **三角函数** (参数为弧度)  
`ASIN(x)`; `COS(x)`; `ACOS(x)`; `TAN(x)`; `ATAN(x)` `ATAN2(x)`; `COT(x)`;

## 字符串函数

---

- **CHAR\_LENGTH(s)**  
字符个数, 汉字算一个;
- **LENGTH(s)**  
同 `CHAR_LENGTH(s)` , 但是汉字算多个字符;
- **CONCAT(s1,s2,...)**、**CONCAT\_WS(x,s1,s2,...)**  
字符串拼接;
- **INSERT(s1,x,len,s2)**  
换对应位置字符串; x 从 1 起始; eg: `SELECT INSERT('12345',1,3,'abc') -- abc45`

- **UPPER(s), UCAASE(S), LOWER(s), LCASE(s)**  
大小写转换;
  - **LEFT(s,n)**  
字符串的前n个字符;
  - **RIGHT(s,n)**  
字符串后n个字符;
  - **LPAD(s1,len,s2)、RPAD(s1,len,s2)**  
s1 长度不够 len 的时候, 在 s1 的左/右填充 s2; ( padding: 填充)
  - **TRIM(s)、LTRIM(s)、RTRIM(s)**  
去除空白字符;
  - **TRIM(s1 FROM s)**  
去除 s 首尾处的 s1 ;
  - **REPEAT(s,n)**  
重复 n 次;
  - **SPACE(n)**  
返回 n 个空格;
  - **REPLACE(s,s1,s2)**  
用 s2 替换 s 中的s1;
  - **STRCMP('abc','abc')**  
字符串比较;
  - **SUBSTRING(s,index,length)、MID(s,n,len)**  
子字符串;
  - **LOCATE(s1,s), POSITION(s1 IN s)**  
s1 在 s 中的位置; ( index 从 1 开始)
  - **INSTR(s,s1) — in string**  
s1 在 s 中的位置;
  - **REVERSE(s)**  
颠倒字符串;
  - **ELT(n,s1,s2,...)**  
Element, 返回第 n 个字符串; eg: `SELECT ELT(2,'a','b','c') -- b`; 参考 `FIELD(s,s1,s2...)`;
  - **EXPORT\_SET(int, s1, s0, separator, n)**
    - 将int的二进制从右向左, 0转为s0, 1转为s1, 并用 separator 作为分隔符拼接为长度为n字符串; eg:  
`select EXPORT_SET(5,'1','0','', 10) — 101000000 \`  
`select EXPORT_SET(5,'Y','N','/', 10) — Y/N/Y/N/N/N/N/N/N/N`
    - n的默认值为64, 最大值也是64 (即使n大于64, 结果长度也是64)
- For every bit of the first argument (which is supplied as an integer but the function works by converting it into bits) it checks whether it is 1 or 0. The order of checking is right to left.
- **FIELD(s,s1,s2...)**  
返回第一个与字符串s匹配的字符串位置; eg: `SELECT FIELD('c','a','b','c') -- 3`

- **FIND\_IN\_SET(s1,s2)**

此函数会将 **s2** 根据 **逗号** 分隔成字符串数组，然后查找数组中 **s1** 匹配的位置，位置排序从 **1** 开始，如果没有匹配的，返回 **0**；  
eg: `select find_in_set('123', 'abcd,1234,efg,123');` — 结果4

- **MAKE\_SET(x,s1,s2 ... )**

- 将x中的数字转为二进制，如： `1 → 1`； `4 → 100`； `31 → 11111`；
- 二进制按照 **低位到高位** 顺序与字符串数组一一对应；
- 取出二进制为1的对应字符串，按原数组顺序，用逗号拼接起来：  
`SELECT MAKE_SET(1|4,'Hello','ABC','MySQL','.', 'XYZ');` — Hello,MySQL； `1|4` 表示 **位与** 运算；

- **SUBSTRING\_INDEX(str, separator, count)**

```
select substring_index('blog.jb51.net', '.', 0) — null
select substring_index('blog.jb51.net', '.', 1) — blog
select substring_index('blog.jb51.net', '.', 2) — blog.jb51
select substring_index('blog.jb51.net', '.', 3) — blog.jb51.net
select substring_index('blog.jb51.net', '.', 4) — blog.jb51.net
count 也可以为负数
```

- **LOAD\_FILE(file\_name)**

加载并显示文件内容；

## 日期时间函数

- **CURDATE(), CURRENT\_DATE()**

- **CURTIME(), CURRENT\_TIME()**

- **NOW(), CURRENT\_TIMESTAMP(), LOCALTIME(), SYSDATE(), LOCALTIMESTAMP()**

## 条件判断函数

- **IF(expr,v1,v2)**

如果 **expr** 成立，返回v1；否则，返回v2；

- **IFNULL(v1,v2)**

如果 **v1** 不为null，返回v1；否则，返回v2；

- **CASE**

```
-- 语法一
SELECT CASE
    WHEN e1
    THEN v1
    WHEN e2
    THEN e2
    ...
    ELSE vn
END
-- 语法二
SELECT CASE 1
    WHEN 1 THEN '我是1'
    WHEN 2 THEN '我是2'
    ELSE '你是谁'
```

## 系统信息函数

- **VERSION()**  
版本；
- **CONNECTION\_ID()**  
用户登录MySQL时，系统分配的连接id
- **DATABASE()、SCHEMA**  
当前使用数据库；
- **USER()、SYSTEM\_USER()、SESSION\_USER()、CURRENT\_USER()、CURRENT\_USER**  
当前用户；
- **CHARSET(str)**  
字符集；如：返回utf8mb4
- **COLLATION(str)**  
返回字符串str的字符排列方式；如：返回utf8mb4\_general\_ci
- **LAST\_INSERT\_ID()**  
返回最近生成的 **AUTO\_INCREMENT** 值；

## 加密函数

- **PASSWORD(str)**  
密码加密函数；
- **MD5(str)**  
求散列值；
- **ENCODE(str,pswd\_str)** 与 **DECODE(crypt\_str,pswd\_str)**  
使用 **pswd\_str** 对 **str** 进行加密/解密

## And So On

- **FORMAT(x,n)**  
格式化；

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

- **进制转化**  
**ASCII(str)** —— 返回str的第一个字符的 **ASCII** 码；  
**BIN(x)** —— 返回x的二进制；  
**HEX(x)** —— 返回x的十六进制；  
**OCT(x)** —— 返回x的八进制；  
**CONV(x,f1,f2)** —— 返回f1进制数变成f2进制数；
- **IP地址函数**  
**ATON** —— address to number  
**NTOA** —— number to address

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

- **GET\_LOCK(name,time)**  
加锁；
- **IS\_FREE\_LOCK(name)**  
锁是否被占用；

- **RELEASE\_LOCK(name)**

释放锁；

- **BENCHMARK(count,expr)**

将表达式**expr**重复执行**count**次；

- **改变字符集**

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

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

- **转换数据类型**

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

```
-- 2个函数只对BINARY、CHAR、DATE、DATETIME、TIME、SIGNED INTEGER、UNSIGNED INTEGER等类型起效；
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)
->2014-12-18
```

总结自：[MySQL函数](#)