

MySQL 函数

MySQL 有很多内置的函数，以下列出了这些函数的说明。

MySQL 字符串函数

函数	描述	实例
ASCII(s)	返回字符串 s 的第一个字符的 ASCII 码。	返回 CustomerName 字段第一个字母的 ASCII 码： <div>SELECT ASCII(CustomerName) AS NumCodeOfFirstChar FROM Customers;</div>
CHAR_LENGTH(s)	返回字符串 s 的字符数	返回字符串 RUNOOB 的字符数 <div>SELECT CHAR_LENGTH("RUNOOB") AS LengthOfString;</div>
CHARACTER_LENGTH(s)	返回字符串 s 的字符数	返回字符串 RUNOOB 的字符数 <div>SELECT CHARACTER_LENGTH("RUNOOB") AS LengthOfString;</div>
CONCAT(s1,s2...sn)	字符串 s1,s2 等多个字符串合并为一个字符串	合并多个字符串 <div>SELECT CONCAT("SQL ", "Runoob ", "Google ", "Facebook") AS ConcatenatedString;</div>
CONCAT_WS(x,s1,s2...sn)	同 CONCAT(s1,s2,...) 函数，但是每个字符串之间要加上 x，x 可以是分隔符	合并多个字符串，并添加分隔符： <div>SELECT CONCAT_WS("-", "SQL", "Tutorial", "is", "fun!")AS ConcatenatedString;</div>
FIELD(s,s1,s2...)	返回第一个字符串 s 在字符串列表 (s1,s2...)中的位置	返回字符串 c 在列表值中的位置：

		<pre>SELECT FIELD("c", "a", "b", "c", "d", "e");</pre>
FIND_IN_SET(s1,s2)	返回在字符串s2中与s1匹配的字符串的位置	<p>返回字符串 c 在指定字符串中的位置：</p> <pre>SELECT FIND_IN_SET("c", "a,b,c,d,e");</pre>
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，最后一位四舍五入。	<p>格式化数字 "#,###.##" 形式：</p> <pre>SELECT FORMAT(250500.563 4, 2); -- 输出 250,5 00.56</pre>
INSERT(s1,x,len,s2)	字符串 s2 替换 s1 的 x 位置开始长度为 len 的字符串	<p>从字符串第一个位置开始的 6 个字符替换为 runoob：</p> <pre>SELECT INSERT("google.co m", 1, 6, "runnob"); -- 输出: runoob.com</pre>
LOCATE(s1,s)	从字符串 s 中获取 s1 的开始位置	<p>获取 b 在字符串 abc 中的位置：</p> <pre>SELECT LOCATE('st','myte ststring'); -- 5</pre> <p>返回字符串 abc 中 b 的位置：</p> <pre>SELECT LOCATE('b', 'abc') -- 2</pre>
LCASE(s)	将字符串 s 的所有字母变成小写字母	<p>字符串 RUNOOB 转换为小写：</p> <pre>SELECT LCASE('RUNOOB') - - runoob</pre>
LEFT(s,n)	返回字符串 s 的前 n 个字符	<p>返回字符串 runoob 中的前两个字符：</p> <pre>SELECT LEFT('runoob',2) -- ru</pre>

LOWER(s)	将字符串 s 的所有字母变成小写字母	字符串 RUNOOB 转换为小写： <pre>SELECT LOWER('RUNOOB') - - runoob</pre>
LPAD(s1,len,s2)	在字符串 s1 的开始处填充字符串 s2，使字符串长度达到 len	将字符串 xx 填充到 abc 字符串的开始处： <pre>SELECT LPAD('abc',5,'xx') -- xxabc</pre>
LTRIM(s)	去掉字符串 s 开始处的空格	去掉字符串 RUNOOB 开始处的空格： <pre>SELECT LTRIM(" RUNOOB") AS LeftTrimmedString ;-- RUNOOB</pre>
MID(s,n,len)	从字符串 s 的 n 位置截取长度为 len 的子字符串，同 SUBSTRING(s,n,len)	从字符串 RUNOOB 中的第 2 个位置截取 3 个字符： <pre>SELECT MID("RUNOOB", 2, 3) AS ExtractString; -- UNO</pre>
POSITION(s1 IN s)	从字符串 s 中获取 s1 的开始位置	返回字符串 abc 中 b 的位置： <pre>SELECT POSITION('b' in 'abc') -- 2</pre>
REPEAT(s,n)	将字符串 s 重复 n 次	将字符串 runoob 重复三次： <pre>SELECT REPEAT('runoob',3) -- runoobrunoobrunoob</pre>
REPLACE(s,s1,s2)	将字符串 s2 替代字符串 s 中的字符串 s1	将字符串 abc 中的字符 a 替换为字符 x： <pre>SELECT REPLACE('abc','a','x') --xbc</pre>
REVERSE(s)	将字符串 s 的顺序反	将字符串 abc 的顺序反过来：

	过来	<pre>SELECT REVERSE('abc') -- cba</pre>
RIGHT(s,n)	返回字符串 s 的后 n 个字符	<p>返回字符串 runoob 的后两个字符：</p> <pre>SELECT RIGHT('runoob',2) -- ob</pre>
RPAD(s1,len,s2)	在字符串 s1 的结尾处添加字符串 s2，使字符串的长度达到 len	<p>将字符串 xx 填充到 abc 字符串的结尾处：</p> <pre>SELECT RPAD('abc',5,'xx') -- abcxx</pre>
RTRIM(s)	去掉字符串 s 结尾处的空格	<p>去掉字符串 RUNOOB 的末尾空格：</p> <pre>SELECT RTRIM("RUNOOB") AS RightTrimmedString ; -- RUNOOB</pre>
SPACE(n)	返回 n 个空格	<p>返回 10 个空格：</p> <pre>SELECT SPACE(10);</pre>
STRCMP(s1,s2)	比较字符串 s1 和 s2，如果 s1 与 s2 相等返回 0，如果 s1>s2 返回 1，如果 s1<s2 返回 -1	<p>比较字符串：</p> <pre>SELECT STRCMP("runoob", "runoob"); -- 0</pre>
SUBSTR(s, start, length)	从字符串 s 的 start 位置截取长度为 length 的子字符串	<p>从字符串 RUNOOB 中的第 2 个位置截取 3 个字符：</p> <pre>SELECT SUBSTR("RUNOOB", 2, 3) AS ExtractString; -- UNO</pre>
SUBSTRING(s, start, length)	从字符串 s 的 start 位置截取长度为 length 的子字符串	<p>从字符串 RUNOOB 中的第 2 个位置截取 3 个字符：</p>

		<pre>SELECT SUBSTRING("RUNOOB", 2, 3) AS ExtractString; -- UNO</pre>
SUBSTRING_INDEX(s, delimiter, number)	<p>返回从字符串 s 的第 number 个出现的分隔符 delimiter 之后的子串。</p> <p>如果 number 是正数，返回第 number 个字符左边的字符串。</p> <p>如果 number 是负数，返回第(number 的绝对值(从右边数))个字符右边的字符串。</p>	<pre>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</pre>
TRIM(s)	去掉字符串 s 开始和结尾处的空格	<p>去掉字符串 RUNOOB 的首尾空格：</p> <pre>SELECT TRIM(' RUNOOB ') AS TrimmedString;</pre>
UCASE(s)	将字符串转换为大写	<p>将字符串 runoob 转换为大写：</p> <pre>SELECT UCASE("runoob"); -- RUNOOB</pre>
UPPER(s)	将字符串转换为大写	<p>将字符串 runoob 转换为大写：</p> <pre>SELECT UPPER("runoob"); -- RUNOOB</pre>

MySQL 数字函数

函数名	描述	实例
ABS(x)	返回 x 的绝对值	<p>返回 -1 的绝对值：</p> <pre>SELECT ABS(-1) -- 返回1</pre>

ACOS(x)	求 x 的反余弦值(参数是弧度)	<pre>SELECT ACOS(0.25);</pre>
ASIN(x)	求反正弦值(参数是弧度)	<pre>SELECT ASIN(0.25);</pre>
ATAN(x)	求反正切值(参数是弧度)	<pre>SELECT ATAN(2.5);</pre>
ATAN2(n, m)	求反正切值(参数是弧度)	<pre>SELECT ATAN2(-0.8, 2);</pre>
AVG(expression)	返回一个表达式的平均值, expression 是一个字段	<p>返回 Products 表中Price 字段的平均值:</p> <pre>SELECT AVG(Price) AS AveragePrice FROM Products;</pre>
CEIL(x)	返回大于或等于 x 的最小整数	<pre>SELECT CEIL(1.5) -- 返回2</pre>
CEILING(x)	返回大于或等于 x 的最小整数	<pre>SELECT CEIL(1.5) -- 返回2</pre>
COS(x)	求余弦值(参数是弧度)	<pre>SELECT COS(2);</pre>
COT(x)	求余切值(参数是弧度)	<pre>SELECT COT(6);</pre>
COUNT(expression)	返回查询的记录总数, expression 参数是一个字段或者 * 号	<p>返回 Products 表中 products 字段总共有多少条记录:</p> <pre>SELECT COUNT(ProductID) AS NumberOfProducts FROM Products;</pre>

DEGREES(x)	将弧度转换为角度	<pre>SELECT DEGREES(3.1415926535898) -- 180</pre>
n DIV m	整除，n 为被除数，m 为除数	<p>计算 10 除以 5：</p> <pre>SELECT 10 DIV 5; -- 2</pre>
EXP(x)	返回 e 的 x 次方	<p>计算 e 的三次方：</p> <pre>SELECT EXP(3) -- 20.085536923188</pre>
FLOOR(x)	返回小于或等于 x 的最大整数	<p>小于或等于 1.5 的整数：</p> <pre>SELECT FLOOR(1.5) -- 返回1</pre>
GREATEST(expr1, expr2, expr3, ...)	返回列表中的最大值	<p>返回以下数字列表中的最大值：</p> <pre>SELECT GREATEST(3, 12, 34, 8, 25); -- 34</pre> <p>返回以下字符串列表中的最大值：</p> <pre>SELECT GREATEST("Google", "Runoob", "Apple"); -- Runoob</pre>
LEAST(expr1, expr2, expr3, ...)	返回列表中的最小值	<p>返回以下数字列表中的最小值：</p> <pre>SELECT LEAST(3, 12, 34, 8, 25); -- 3</pre> <p>返回以下字符串列表中的最小值：</p> <pre>SELECT LEAST("Google", "Runoob", "Apple"); -- Apple</pre>

LN	返回数字的自然对数	返回 2 的自然对数： <pre>SELECT LN(2); -- 0.6931471805599453</pre>
LOG(x)	返回自然对数(以 e 为底的对数)	<pre>SELECT LOG(20.08553 6923188) -- 3</pre>
LOG10(x)	返回以 10 为底的对数	<pre>SELECT LOG10(100) - - 2</pre>
LOG2(x)	返回以 2 为底的对数	返回以 2 为底 6 的对数： <pre>SELECT LOG2(6); -- 2.584962500721156</pre>
MAX(expression)	返回字段 expression 中的最大值	返回数据表 Products 中字段 Price 的最大值： <pre>SELECT MAX(Price) A S LargestPrice FROM Products;</pre>
MIN(expression)	返回字段 expression 中的最小值	返回数据表 Products 中字段 Price 的最小值： <pre>SELECT MIN(Price) A S LargestPrice FROM Products;</pre>
MOD(x,y)	返回 x 除以 y 以后的余数	5 除以 2 的余数： <pre>SELECT MOD(5,2) -- 1</pre>
PI()	返回圆周率(3.141593)	<pre>SELECT PI() --3.141 593</pre>

POW(x,y)	返回 x 的 y 次方	2 的 3 次方： <pre>SELECT POW(2,3) -- 8</pre>
POWER(x,y)	返回 x 的 y 次方	2 的 3 次方： <pre>SELECT POWER(2,3) - - 8</pre>
RADIANS(x)	将角度转换为弧度	180 度转换为弧度： <pre>SELECT RADIANS(180) -- 3.1415926535898</pre>
RAND()	返回 0 到 1 的随机数	<pre>SELECT RAND() --0.9 3099315644334</pre>
ROUND(x)	返回离 x 最近的整数	<pre>SELECT ROUND(1.2345 6) --1</pre>
SIGN(x)	返回 x 的符号，x 是负数、0、正数分别返回 -1、0 和 1	<pre>SELECT SIGN(-10) -- (-1)</pre>
SIN(x)	求正弦值(参数是弧度)	<pre>SELECT SIN(RADIANS(30)) -- 0.5</pre>
SQRT(x)	返回x的平方根	25 的平方根： <pre>SELECT SQRT(25) -- 5</pre>
SUM(expression)	返回指定字段的总和	计算 OrderDetails 表中字段 Quantity 的总和： <pre>SELECT SUM(Quantity) AS TotalItemsOrde red FROM OrderDetai ls;</pre>

TAN(x)	求正切值(参数是弧度)	<pre>SELECT TAN(1.75); -- -5.5203799225093 3</pre>
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）	<pre>SELECT TRUNCATE(1.2 3456,3) -- 1.234</pre>

MySQL 日期函数

函数名	描述	实例
ADDDATE(d,n)	计算起始日期 d 加上 n 天的日期	<pre>SELECT ADDDATE("20 17-06-15", INTERVA L 10 DAY); ->2017-06-25</pre>
ADDTIME(t,n)	时间 t 加上 n 秒的时间	<pre>SELECT ADDTIME('20 11-11-11 11:11:11' , 5) ->2011-11-11 11:11 :16 (秒)</pre>
CURDATE()	返回当前日期	<pre>SELECT CURDATE(); -> 2018-09-19</pre>
CURRENT_DATE()	返回当前日期	<pre>SELECT CURRENT_DAT E(); -> 2018-09-19</pre>
CURRENT_TIME	返回当前时间	<pre>SELECT CURRENT_TIM E(); -> 19:59:02</pre>
CURRENT_TIMESTAMP()	返回当前日期和时间	<pre>SELECT CURRENT_TIM ESTAMP()</pre>

		<pre>-> 2018-09-19 20:57:43</pre>
CURTIME()	返回当前时间	<pre>SELECT CURTIME(); -> 19:59:02</pre>
DATE()	从日期或日期时间表达式中提取日期值	<pre>SELECT DATE("2017-06-15"); -> 2017-06-15</pre>
DATEDIFF(d1,d2)	计算日期 d1->d2 之间相隔的天数	<pre>SELECT DATEDIFF('2001-01-01','2001-02-02') -> -32</pre>
DATE_ADD(d, INTERVAL expr type)	计算起始日期 d 加上一个时间段后的日期	<pre>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)</pre>
DATE_FORMAT(d,f)	按表达式 f 的要求显示日期 d	<pre>SELECT DATE_FORMAT('2011-11-11 11:11:11','%Y-%m-%d %r') -> 2011-11-11 11:11 AM</pre>
DATE_SUB(date,INTERVAL expr type)	函数从日期减去指定的时间间隔。	<p>Orders 表中 OrderDate 字段减去 2 天:</p> <pre>SELECT OrderId,DATE_SUB(OrderDate,IN</pre>

		Terval 2 DAY) AS 0 rderPayDate FROM Orders
DAY(d)	返回日期值 d 的日期部分	<pre>SELECT DAY("2017-06-15"); -> 15</pre>
DAYNAME(d)	返回日期 d 是星期几，如 Monday,Tuesday	<pre>SELECT DAYNAME('2011-11-11 11:11:11') ->Friday</pre>
DAYOFMONTH(d)	计算日期 d 是本月的第几天	<pre>SELECT DAYOFMONTH('2011-11-11 11:11:11') ->11</pre>
DAYOFWEEK(d)	日期 d 今天是星期几，1 星期日，2 星期一，以此类推	<pre>SELECT DAYOFWEEK('2011-11-11 11:11:11') ->6</pre>
DAYOFYEAR(d)	计算日期 d 是本年的第几天	<pre>SELECT DAYOFYEAR('2011-11-11 11:11:11') ->315</pre>
EXTRACT(type FROM d)	从日期 d 中获取指定的值，type 指定返回的值。 type可取值为： MICROSECOND SECOND MINUTE HOUR DAY WEEK MONTH	<pre>SELECT EXTRACT(MINUTE FROM '2011-11-11 11:11:11') -> 11</pre>

	QUARTER YEAR SECOND_MICROSECOND MINUTE_MICROSECOND MINUTE_SECOND HOUR_MICROSECOND HOUR_SECOND HOUR_MINUTE DAY_MICROSECOND DAY_SECOND DAY_MINUTE DAY_HOUR YEAR_MONTH	
FROM_DAYS(n)	计算从 0000 年 1 月 1 日开始 n 天后的日期	<pre>SELECT FROM_DAYS(111) -> 0003-01-16</pre>
HOUR(t)	返回 t 中的小时值	<pre>SELECT HOUR('1:2:3') -> 1</pre>
LAST_DAY(d)	返回给给定日期的那一月份的最后一天	<pre>SELECT LAST_DAY("2017-06-20"); -> 2017-06-30</pre>
LOCALTIME()	返回当前日期和时间	<pre>SELECT LOCALTIME() -> 2018-09-19 20:57:43</pre>
LOCALTIMESTAMP()	返回当前日期和时间	<pre>SELECT LOCALTIMESTAMP() -> 2018-09-19 20:57:43</pre>

MAKEDATE(year, day-of-year)	基于给定参数年份 year 和所在年中的天数序号 day-of-year 返回一个日期	<pre>SELECT MAKEDATE(2017, 3); -> 2017-01-03</pre>
MAKETIME(hour, minute, second)	组合时间，参数分别为小时、分钟、秒	<pre>SELECT MAKETIME(11, 35, 4); -> 11:35:04</pre>
MICROSECOND(date)	返回日期参数所对应的微秒数	<pre>SELECT MICROSECOND("2017-06-20 09:34:00.000023"); -> 23</pre>
MINUTE(t)	返回 t 中的分钟值	<pre>SELECT MINUTE('1:2:3'); -> 2</pre>
MONTHNAME(d)	返回日期当中的月份名称，如 November	<pre>SELECT MONTHNAME('2011-11-11 11:11:11'); -> November</pre>
MONTH(d)	返回日期d中的月份值，1 到 12	<pre>SELECT MONTH('2011-11-11 11:11:11'); ->11</pre>
NOW()	返回当前日期和时间	<pre>SELECT NOW(); -> 2018-09-19 20:57:43</pre>
PERIOD_ADD(period, number)	为 年-月 组合日期添加一个时段	<pre>SELECT PERIOD_ADD(201703, 5); -> 201708</pre>
PERIOD_DIFF(period1,	返回两个时段之间的月份	

period2)	差值	<pre>SELECT PERIOD_DIFF (201710, 201703); -> 7</pre>
QUARTER(d)	返回日期d是第几季节, 返回 1 到 4	<pre>SELECT QUARTER('20 11-11-11 11:11:11') -> 4</pre>
SECOND(t)	返回 t 中的秒钟值	<pre>SELECT SECOND('1: 2:3') -> 3</pre>
SEC_TO_TIME(s)	将以秒为单位的时间 s 转 换为时分秒的格式	<pre>SELECT SEC_TO_TIME (4320) -> 01:12:00</pre>
STR_TO_DATE(string, format_mask)	将字符串转变为日期	<pre>SELECT STR_TO_DATE ("August 10 2017", "%M %d %Y"); -> 2017-08-10</pre>
SUBDATE(d,n)	日期 d 减去 n 天后的日期	<pre>SELECT SUBDATE('20 11-11-11 11:11:11' , 1) ->2011-11-10 11:11 :11 (默认是天)</pre>
SUBTIME(t,n)	时间 t 减去 n 秒的时间	<pre>SELECT SUBTIME('20 11-11-11 11:11:11' , 5) ->2011-11-11 11:11 :06 (秒)</pre>
SYSDATE()	返回当前日期和时间	<pre>SELECT SYSDATE() -> 2018-09-19 20:5 7:43</pre>

TIME(expression)	提取传入表达式的时间部分	<pre>SELECT TIME("19:30:10"); -> 19:30:10</pre>
TIME_FORMAT(t,f)	按表达式 f 的要求显示时间 t	<pre>SELECT TIME_FORMAT('11:11:11','%r') 11:11:11 AM</pre>
TIME_TO_SEC(t)	将时间 t 转换为秒	<pre>SELECT TIME_TO_SEC('1:12:00') -> 4320</pre>
TIMEDIFF(time1, time2)	计算时间差值	<pre>SELECT TIMEDIFF("13:10:11", "13:10:10"); -> 00:00:01</pre>
TIMESTAMP(expression, interval)	单个参数时，函数返回日期或日期时间表达式；有2个参数时，将参数加和	<pre>SELECT TIMESTAMP("2017-07-23", "13:10:11"); -> 2017-07-23 13:10:11</pre>
TO_DAYS(d)	计算日期 d 距离 0000 年 1 月 1 日的天数	<pre>SELECT TO_DAYS('0001-01-01 01:01:01') -> 366</pre>
WEEK(d)	计算日期 d 是本年的第几个星期，范围是 0 到 53	<pre>SELECT WEEK('2011-11-11 11:11:11') -> 45</pre>
WEEKDAY(d)	日期 d 是星期几，0 表示星期一，1 表示星期二	<pre>SELECT WEEKDAY("2017-06-15"); -> 3</pre>
WEEKOFYEAR(d)	计算日期 d 是本年的第几	

	个星期，范围是 0 到 53	<pre>SELECT WEEKOFYEAR('2011-11-11 11:11: 11') -> 45</pre>
YEAR(d)	返回年份	<pre>SELECT YEAR("2017- 06-15"); -> 2017</pre>
YEARWEEK(date, mode)	返回年份及第几周（0到53），mode 中 0 表示周天，1表示周一，以此类推	<pre>SELECT YEARWEEK("2 017-06-15"); -> 201724</pre>

MySQL 高级函数

函数名	描述	实例
BIN(x)	返回 x 的二进制编码	15 的 2 进制编码: <pre>SELECT BIN (15); -- 1 111</pre>
BINARY(s)	将字符串 s 转换为二进制字符串	<pre>SELECT BIN ARY "RUNOO B"; -> RUNOOB</pre>
<pre>CASE express ion WHEN con dition1 THEN result1 WHEN con dition2 THEN result2 ... WHEN con ditionN THEN resultN ELSE res ult END</pre>	CASE 表示函数开始，END 表示函数结束。如果 condition1 成立，则返回 result1，如果 condition2 成立，则返回 result2，当全部不成立则返回 result，而当有一个成立之后，后面的就不执行了。	<pre>SELECT CAS E WHEN 1 > 0 THEN '1 > 0' WHEN 2 > 0 THEN '2 > 0' ELSE '3 > 0' END ->1 > 0</pre>

CAST(x AS type)	转换数据类型	字符串日期转换为日期： <pre>SELECT CAST("2017-08-29" AS DATE); -> 2017-08-29</pre>
COALESCE(expr1, expr2, ..., expr_n)	返回参数中的第一个非空表达式（从左向右）	<pre>SELECT COALESCE(NULL, NULL, NULL, 'runoob.com', NULL, 'google.com'); -> runoob.com</pre>
CONNECTION_ID()	返回服务器的连接数	<pre>SELECT CONNECTION_ID(); -> 4292835</pre>
CONV(x,f1,f2)	返回 f1 进制数变成 f2 进制数	<pre>SELECT CONV(15, 10, 2); -> 1111</pre>
CONVERT(s USING cs)	函数将字符串 s 的字符集变成 cs	<pre>SELECT CHARSET('ABC') -> utf-8 SELECT CHARSET(CONVERT('ABC' USING gbk)) -> gbk</pre>
CURRENT_USER()	返回当前用户	<pre>SELECT CURRENT_USER</pre>

		<pre>(); -> guest@%</pre>
DATABASE()	返回当前数据库名	<pre>SELECT DAT ABASE(); -> runoob</pre>
IF(expr,v1,v2)	如果表达式 expr 成立，返回结果 v1；否则，返回结果 v2。	<pre>SELECT IF(1 > 0, '正 确', '错误') ->正确</pre>
<u>IFNULL(v1,v2)</u>	如果 v1 的值不为 NULL，则返回 v1，否则返回 v2。	<pre>SELECT IFN ULL(null, 'Hello Wor d') ->Hello Wo rd</pre>
ISNULL(expression)	判断表达式是否为 NULL	<pre>SELECT ISN ULL(NULL); ->1</pre>
LAST_INSERT_ID()	返回最近生成的 AUTO_INCREMENT 值	<pre>SELECT LAS T_INSERT_I D(); ->6</pre>
NULLIF(expr1, expr2)	比较两个字符串，如果字符串 expr1 与 expr2 相等 返回 NULL，否则返回 expr1	<pre>SELECT NUL LIF(25, 25); -></pre>
SESSION_USER()	返回当前用户	<pre>SELECT SES SION_USER (); -> guest@%</pre>

SYSTEM_USER()	返回当前用户	<pre>SELECT SYSTEM_USER (); -> guest@%</pre>
USER()	返回当前用户	<pre>SELECT USER (); -> guest@%</pre>
VERSION()	返回数据库的版本号	<pre>SELECT VERSION() -> 5.6.34</pre>

[← MySQL 导入数据](#)

[MySQL 运算符 →](#)