

# 演算子 / 関数

## SELECT と WHERE 節で使用する関数

### グループ化

( ) 式の評価順の変更。

### 算術演算

+ 足し算。  
- 引き算。  
\* 掛け算。  
/ 割り算。

### ビット演算

| ビット演算の和。  
& ビット演算の積。  
<< 左ビットシフト。  
>> 右ビットシフト。  
~ ビット反転。  
BIT\_COUNT (n) 数値 n のビット 1 の数。

### 論理演算

NOT、! 否定。  
OR、| | 論理和。  
AND、& & 論理積。

### 比較演算

= 等しい。  
< >、! = 等しくない。  
< = 以下。  
< より小さい。  
> = 以上。  
> より大きい。  
< = > 等しい ( NULL 同士でも成立 )。

IS NULL、IS NOT NULL

NULLかどうかの判定。

BETWEEN a AND b

値が a と b の間にあれば真。

IN (a、b、.....)、NOT IN (a、b、.....)

リスト中に値があるかどうか。

ISNULL (a)

a が NULL なら真。

COALESCE (a、b、.....)

リスト中の最初の NULL でない値。

INTERVAL (n、n1、n2、.....)

n < n1 なら 0、n < n2 なら 1、.....を返す。

## 文字列比較

LIKE、NOT LIKE

簡易正規表現での文字列比較。%は任意の文字列に、  
\_は任意の一文字に適合する。

REGEXP、RLIKE、NOT、REGEXP、NOT RLIKE

正規表現での文字列比較。

STRCMP (a、b)

文字列 a と b が等しければ 0。a が小さければ - 1、  
b が小さければ 1。

## キャスト

BINARY

バイナリへのキャスト。

## フロー制御

IFNULL

a が NULL でなければ a、a が NULL であれば b。

NULLIF

a = b なら a、a != b なら NULL。

IF (a、b、c)

a が真なら b、偽なら c。

CASE a WHEN b THEN c WHEN d THEN e ..... ELSE f END

a が b であれば c、d であれば e、.....、どれでもなければ f を返す。

CASE WHEN a THEN b WHEN c THEN d ..... ELSE e END

a が真であれば b、c が真であれば d、.....どれも成り立たなければ e を返す。

## 数学関数

-	符号変換。
ABS ( x )	絶対値。
SIGN ( x )	符号。
MOD ( n, m ), n%m	剰余。n を m で割った余り。
FLOOR ( x )	x 以下の最大の整数。
CEILING ( x )	x 以上の最小の整数。
ROUND ( x )	x の小数点以下を四捨五入した整数。
ROUND ( x, d )	x を小数点以下 d + 1 桁で四捨五入した値。
EXP ( x )	指数関数。
LOG ( x )	自然対数。
LOG10 ( x )	常用対数。
POW ( x, y ), POWER ( x, y )	x の y 乗。
SQRT ( x )	平方根。
PI ( )	。
COS ( x )	コサイン。
SIN ( x )	サイン。
TAN ( x )	タンジェント。
ACOS ( x )	アークコサイン。
ASIN ( x )	アークサイン。
ATAN ( x )	アークタンジェント。
ATAN2 ( x, y )	y / x のアークタンジェント。
COT ( x )	コタンジェント。
RAND( ), RAND(n)	0 ~ 1.0 の間の乱数。n の指定は乱数の種の設定。
LEAST ( x, y, ..... )	リスト中の最小値。
GREATEST ( x, y, ..... )	リスト中の最大値。
DEGRESS ( x )	ラジアンから度に変換。
RADIANS ( x )	度からラジアンに変換。
TRUNCATE ( x, d )	x を小数点以下 d + 1 桁で切捨て。

## 文字列関数

ASCII ( s )	s の最初の文字の ASCII コード。
ORD ( s )	s の最初の文字の ASCII コード ( マルチバイト対応 )
CONV ( n, f, t )	n を f 進数から t 進数に変換。
BIN ( n )	2 進数に変換。
OCT ( n )	8 進数に変換。
HEX ( n )	16 進数に変換。
CHAR ( n, ..... )	リストを ASCII コードとみなして得られる文字列。
CONCAT ( s1, s2, ..... )	文字列の結合。
CONCAT_WS ( sep, s1, s2, ..... )	sep を区切り文字として文字列を結合。
LENGTH ( s ), OCTET_LENGTH ( s )	文字列のバイト数。
CHAR_LENGTH ( s ), CHARACTER_LENGTH ( s )	文字列の文字数 ( マルチバイト対応 )
LOCATE ( sub, s ), POSITION ( sub IN s ), INSTR ( s, sub )	文字列 s 中で sub が現れる位置 ( マルチバイト対応 )
LOCATE ( sub, s, p )	文字列 s 中の位置 p 以降で sub が現れる位置 ( マルチバイト対応 )
LPAD ( s, 1, p )	長さが 1 になるまで文字列 s の左に p を埋める。
RPAD ( s, 1, p )	長さが 1 になるまで文字列 s の右に p を埋める。
LEFT ( s, 1 )	文字列 s の左 1 文字 ( マルチバイト対応 )
RIGHT ( s, 1 )	文字列 s の右 1 文字 ( マルチバイト対応 )
SUBSTRING ( s, p, 1 ), SUBSTRING ( s FROM p FOR 1 ), MID ( s, p, 1 )	文字列 s の位置 p から 1 文字。
SUBSTRING ( s, p ), SUBSTRING ( s FROM p )	文字列 s の位置 p 以降の文字列。

SUBSTRING_INDEX ( s, d, c )	文字列 s を d で区切った左から c 個分の文字列。 c が負の場合は右から。
LTRIM ( s )	文字列の左の空白文字を削除。
RTRIM ( s )	文字列の右の空白文字を削除。
TRIM ( [ [ BOTH   LEADING   TRAILING ] [ r ] FROM ] s )	文字列 s から文字 r を削除。 r が与えられない場合は空白文字。 LEADING は文字列の先頭から、 TRAILING は文字列の最後から、 BOTH は両方から削除する。
SOUNDEX ( s )	文字列 s の soundex 文字列。
SPACE ( n )	n 個の空白文字。
REPLACE ( s, f, t )	文字列 s 中の文字列 f を t に置換 (マルチバイト対応 )
REPEAT ( s, c )	文字列 s の c 回繰り返し。
REVERSE ( s )	文字列の反転 (マルチバイト対応 )
INSERT ( s, p, 1, n )	文字列 s の位置 p から 1 文字を n に置換 (マルチバイト対応 )
ELT ( n, s1, s2, ..... )	n が 1 の場合 s1、 2 の場合 s2、 ..... を返す。
FIELD ( s, s1, s2, ..... )	s が s1 と等しければ 1, s2 と等しければ 2, ..... を返す。
FIND_IN_SET ( s, 1 )	文字列 s の 1 中の位置。 1 は、で区切られた文字列のリスト。
MAKE_SET ( b, s1, s2, ..... )	ビット b に対応する文字列のリスト。
EXPORT_SET ( b, on, off [ , sep [ , n ] ] )	数値 b のビット 1 を文字列 on、ビット 0 を文字列 off にして、 sep (デフォルトで、) 区切った n 個のリスト。
LCASE ( s ), LOWER ( s )	文字列を小文字に変換 (マルチバイト対応 )
UCASE ( s ), UPPER ( s )	文字列を大文字に変換 (マルチバイト対応 )
LOAD_FILE ( f )	ファイル f の内容。

## 日付と時刻関数

DAYOFWEEK ( d )	曜日を表す数値 ( 1 = 日曜日、 2 = 月曜日、 ..... )。
WEEKDAY ( d )	曜日を表す数値 ( 0 = 日曜日、 1 = 月曜日、 ..... )。
DAYOFMONTH ( d )	月の日 ( 1 ~ 31 )。
DAYOFYEAR ( d )	年の日 ( 1 ~ 366 )。
MONTH ( d )	月 ( 1 ~ 12 )。
DAYNAME ( d )	曜日の名前。
MONTHNAME ( d )	月の名前。
QUARTER ( d )	四半期 ( 1 ~ 4 )。
WEEK(d)、WEEK(d、 f)	年の週( 1 ~ 53 )、f が 0 の場合は日曜日が週の始め、1 の場合は月曜日が週の始め。
YEAR ( d )	年 ( 1000 ~ 9999 )。
YEARWEEK(d)、YEARWEEK(d、 f)	年と年の週。f が 0 の場合は日曜日が週の始め、1 の場合は月曜日が週の始め。
HOUR ( t )	時 ( 0 ~ 23 )。
MINUTE ( t )	分 ( 0 ~ 59 )。
SECOND ( t )	秒 ( 0 ~ 59 )。
PERIOD_ADD(p、 n)	期間 p に n 月を追加。
PERIOD_DIFF(p1、 p2)	期間 p1 と p2 の差の月。
DATE_ADD ( d、 INTERVAL e t )、 DATE_SUB ( d、 INTERVAL e t )、 ADDDATE ( d、 INTERVAL e t )、 SUBDATE ( d、 INTERVAL e t )	日付の演算。e は数値で、t は e の単位。

t	意味	E の形式
SECOND	秒	数値
MINUTE	分	数値
HOUR	時	数値
DAY	日	数値
MONTH	月	数値
YEAR	年	数値
MITUTE_SECOND	分と秒	“ mm:ss ”
HOUR_MINUTE	時と分	“ hh:mm ”
DAY_HOUR	日と時	“ dd hh ”

YEAR_MONTH	時と月	“ yy-mm ”
HOUR_SECOND	時、分、秒	“ hh:mm:ss ”
DAY_MINUTE	日、時、分	“ dd hh:mm ”
DAY_SECOND	日、時、分、秒	“ dd hh:mm:ss ”

TO\_DAYS ( d )                      0年からの日数。  
FROM\_DAYS ( n )                    nを0年からの日数とした日付。  
DATE\_FORMAT ( d、 f )            日付 d を、形式 f で整形。

書式指定	意味
%M	月名
%W	曜日名
%D	月の日 ( サフィックス付き )
%Y	4桁の年
%y	2桁の年
%X	週の最初を日曜日とした週の年
%x	週の最初を月曜日とした週の年
%a	曜日名 ( 省略形 )
%d	月の日 ( 00 ~ 31 )
%e	月の日 ( 0 ~ 31 )
%m	月 ( 01 ~ 12 )
%c	月 ( 1 ~ 12 )
%b	月名 ( 省略形 )
%j	年の日 ( 001 ~ 366 )
%H	時 ( 00 ~ 23 )
%k	時 ( 0 ~ 23 )
%h	時 ( 01 ~ 12 )
%l	時 ( 01 ~ 12 )
%I	時 ( 1 ~ 23 )
%i	分 ( 00 ~ 59 )
%r	12時間制時刻 ( hh : mm : ss [ AP ] M )
%T	24時間制時刻 ( hh : mm : ss )
%S	秒 ( 00 ~ 59 )

<b>%s</b>	秒 ( 00 ~ 59 )
<b>%p</b>	AM または PM
<b>%w</b>	週の日 (0=日曜日、1=月曜日、.....)
<b>%U</b>	週 (0 ~ 53) 週の始まりは日曜日
<b>%u</b>	週 (0 ~ 53) 週の始まりは月曜日
<b>%V</b>	週 (1 ~ 53) 週の始まりは日曜日
<b>%v</b>	週 (1 ~ 53) 週の始まりは月曜日
<b>%%</b>	文字%

TIME\_FORMAT ( t, f ) 時、分、秒に限った DATE\_FORMAT。

CURDATE ( )、CURRENT\_DATE ( )

現在の日付。

CURTIME ( )、CURRENT\_TIME ( )

現在の時刻。

NOW ( )、SYSDATE ( )、CURRENT\_TIMESTAMP ( )

現在の日時。

UNIX\_TIMESTAMP ( )、UNIX\_TIMESTAMP ( d )

UNIX タイムスタンプ ( 1970-01-01 00:00:00GMT からの秒数 )。

FROM\_UNIXTIME ( t )

UNIX タイムスタンプ t をタイムスタンプ文字列に変換。

FROM\_UNIXTIME ( t, f )

UNIX タイムスタンプ t を書式 f の文字列に変換。f は DATE\_FORMAT ( ) と同様。

SEC\_TO\_TIME ( s )

秒 s を時刻に変換。

TIME\_TO\_SEC ( t )

時間 t を秒に変換。

## その他の関数

COUNT ( e )	レコード数。
COUNT(DISTINCT e[, e、 .....])	異なる値のレコード数。
AVG ( e )	平均値。
MIN ( e )	最小値。
MAX ( e )	最大値。
SUM ( e )	合計値。
STD ( e )、 STDDEV ( e )	標準誘導。
BIT_OR ( e )	論理和。
BIT_AND ( e )	論理積。