

付録1 データ型と関数

付録1 データ型と関数

データ型

データ型

つぎのような型があります。

- 数値型
- 浮動小数点型
- 日付型
- 文字列型
- バイナリデータ型
- ラージオブジェクト型
- 論理データ型

名称	説明
TINYINT	1 バイトの整数
SMALLINT	2 バイトの整数
MEDIUMINT	3 バイトの整数
INTEGER、INT	4 バイトの整数
BIGINT	8 バイトの整数
NUMERIC、DECIMAL、DEC	パック 10 進数
FLOAT	浮動小数点数
DOUBLE	浮動小数点数
DOUBLE PRECISION、REAL	倍精度浮動小数点
CHAR	固定長文字列
VARCHAR	可変長文字列
TINYTEXT	255 バイト可変長文字列
TEXT	65,535 バイト可変長文字列
MEDIUMTEXT	16,777,215 バイト可変長文字列
LONGTEXT	4,294,967,295 バイト可変長文字列
DATE	日付
DATETIME	日付、時刻
TIMESTAMP	日付、時刻
TIME	時刻
YEAR	年
TINYBLOB	255 バイトバイナリデータ
BLOB	65,535 バイトバイナリデータ
MEDIUMBLOB	16,777,215 バイトバイナリデータ
LOB	4,294,967,295 バイトバイナリデータ

数値型

型名	格納サイズ	範囲
TINYINT	1 バイトの整数	-128 から 127 (符号無しの場合 0 から 255)
SMALLINT	2 バイトの整数	-32768 から 32767 (符号無しの場合 0 から 65535)
MEDIUMINT	3 バイトの整数	-8388608 から 8388607 (符号無しの場合 0 から 16777215)
INTEGER、INT	4 バイトの整数	-2147483648 から 2147483647 (符号無しの場合 0 から 4294967295)
BIGINT	8 バイトの整数	-9223372036854775808 から 9223372036854775807 (符号無しの場合 0 から 18446744073709551615)
NUMERIC、DECIMAL、DEC	パック 10 進数	可変

浮動小数点型

型名	格納サイズ	範囲
FLOAT	浮動小数点数	-3.402823466E+38 から -1.175494351E-38 0 1.175494351E-38 から 3.402823466E+38
DOUBLE	浮動小数点数	-1.7976931348623157E+308 から -2.2250738585072014E-308 0 2.2250738585072014E-308 から 1.7976931348623157E+308
DOUBLE PRECISION、REAL	倍精度浮動小数点	DOUBLE と同様

文字列型

固定長文字列

格納するバイト数を固定した文字列型。格納する値が指定されたバイト数に満たない場合は、指定されたバイト数になるように右側に半角スペースで埋められる。

可変長文字列

格納するバイト数が固定でない(可変な)文字列型

例題

固定長文字列 char と、可変長文字列 varchar の長さを比較します。つぎの SQL スクリプトを実行してください。

```
> DROP TABLE IF EXISTS typetesting01;

> CREATE TABLE typetesting01 (
    c      CHAR(10)
  , v      VARCHAR(10)
);

> INSERT INTO typetesting01 VALUES('Hello', 'Hello');
> INSERT INTO typetesting01 VALUES('Hello World', 'Hello World');
> INSERT INTO typetesting01 VALUES('はろー', 'はろー');

> SELECT * FROM typetesting01;

> SELECT 'length', c, length(c), length(v) FROM typetesting01
UNION
SELECT 'character_length', c, character_length(c), character_length(v) FROM
typetesting01
UNION
SELECT 'octet_length', c, octet_length(c), octet_length(v) FROM typetesting01;
```

結果表

length	c	length(c)	length(v)
length	Hello	10	5
length	はろー	10	3
character_length	Hello	10	5
character_length	はろー	10	3
octet_length	Hello	10	5
octet_length	はろー	13	6

6 rows in set (0.00 sec)

日付 / 時刻型

型	格納サイズ	説明	範囲
DATE	3 バイト	日付	1000-01-01 から 9999-12-31
DATETIME	8 バイト	日付と時刻両方	1000-01-01 00:00:00 から 9999-12-31 23:59:59
TIMESTAMP	4 バイト	タイムスタンプ	1970-01-01 00:00:01 から 2038-01-19 03:14:07
TIME	3 バイト	時刻のみ	-838:59:59 から 838:59:59
YEAR	1 バイト	年	4 桁の場合 : 1901 から 2155

日付の演算

"日付 ± INTERVAL 数値" で、何日後 / 前の日付を計算できます。

```
> SELECT NOW() + INTERVAL 100 DAY;
```

```
    NOW() + INTERVAL 100 DAY
```

```
-----
2016-06-23 19:16:45
```

```
> SELECT NOW() - INTERVAL 100 DAY;
```

```
    NOW() - INTERVAL 100 DAY
```

```
-----
2015-12-06 19:19:14
```

"DATEDIFF"で、日付間の日数を計算できます。

```
> SELECT DATEDIFF(NOW(), '2010-01-01');
```

```
DATEDIFF(NOW(), '2010-01-01')
```

```
-----
                2265
```

```
> SELECT DATEDIFF(NOW(), '2100-01-01');
```

```
DATEDIFF(NOW(), '2010-01-01')
```

```
-----
               -30607
```

ラージ・オブジェクト

型名	格納サイズ	説明
TINYBLOB	255 バイト以下	-
BLOB	65,535 バイト(64KB)以下	-
MEDIUMBLOB	16,777,215 バイト(16MB)以下	-
LONGBLOB	4,294,967,295 バイト (4GB) 以下	-

論理値データ型

標準 SQL の boolean 型が提供されています。boolean は"真"または"偽"という 2 つしかない値のどちらかを取ることができますが MySQL では TINYINT を使って真 (1)、偽 (0) を表現しています。

第 3 の状態である"不明"は SQL の null 値で表現されます。

関数

算術関数

関数	説明	例
<code>abs(x)</code>	絶対値	<code>abs(-17.4)</code>
<code>ceil({dp numeric})</code>	引数より小さくない最小の整数	<code>ceil(-42.8)</code>
<code>degrees(dp)</code>	度に対応するラジアン	<code>degrees(0.5)</code>
<code>exp({dp numeric})</code>	指数	<code>exp(1.0)</code>
<code>floor({dp numeric})</code>	引数より大きくない最大の整数	<code>floor(-42.8)</code>
<code>ln({dp numeric})</code>	自然対数	<code>ln(2.0)</code>
<code>log({dp numeric})</code>	10 を底とした対数(常用対数)	<code>log(100.0)</code>
<code>log(b numeric, x numeric)</code>	底に対する対数 b	<code>log(2.0, 64.0)</code>
<code>mod(y, x)</code>	y/x の剰余	<code>mod(9,4)</code>
<code>pi()</code>	"円周率(π)" 定数	<code>pi()</code>
<code>pow(x numeric, e numeric)</code>	e 乗	<code>pow(9.0, 3.0)</code>
<code>radians(dp)</code>	ラジアンに対応する度	<code>radians(45.0)</code>
<code>rand()</code>	0.0 ~ 1.0 の乱数値	<code>rand()</code>
<code>round(v numeric, s integer)</code>	小数点位置 s で四捨五入	<code>round(42.4382, 2)</code>
<code>sign(numeric or numeric)</code>	引数の符号 (-1, 0, +1)	<code>sign(-8.4)</code>
<code>sqrt(dp or numeric)</code>	平方根	<code>sqrt(2.0)</code>
<code>truncate(v numeric, s integer)</code>	小数点位置 s で切捨て	<code>truncate(42.4382, 2)</code>

文字列関数

関数	説明	例
<code>concat(str, str[,str])</code>	文字列結合	<code>concat('My','SQL')</code>
<code>bit_length(str)</code>	文字列のビット数	<code>bit_length('jose')</code>
<code>char_length(str)</code> or <code>character_length(str)</code>	文字列の中の文字数	<code>char_length('jose')</code>
<code>lower(str)</code>	文字列を小文字に変換	<code>lower('TOM')</code>
<code>octet_length(str)</code>	文字列のバイト数	<code>octet_length('jose')</code>
<code>insert(str,pos,len,newstr)</code>	部分文字列の挿入	<code>overlay('Quadratic', 3, 4, 'What')</code>
<code>position(substr in str)</code>	指定された部分文字列の場所	<code>position('om' in 'Thomas')</code>
<code>trim([leading trailing both] [characters] from string)</code>	characters（デフォルトではスペース）で指定された文字のみを含む最も長い文字列を、string の最初、最後、そして両側から削除	<code>trim(both 'x' from 'xTomxx')</code>
<code>substr(str, pos)</code>	部分文字列の取り出し	<code>substring('Thomas' ,3)</code>
<code>substr(str, pos, len)</code>	部分文字列の取り出し（指定部分から指定文字数）	<code>substring('Thomas' 1, 4)</code>
<code>upper(string)</code>	文字列を大文字に変換	<code>upper('tom')</code>
<code>length(string)</code>	文字列の長さ	<code>length('jose')</code>
<code>repeat(text, integer)</code>	テキストを指定回数分繰り返す	<code>repeat('Pg', 4)</code>
<code>replace(string text, fromtext, totext)</code>	string に出現するすべての from 部分文字列を to 部分文字列に置換	<code>replace('abcdefabcdef', 'cd', 'XX')</code>
<code>unhex(number_str)</code>	number を、同等の 16 進数表現に変換	<code>unhex('4D7953514C')</code>

フォーマット関数

関数	説明	例
<code>format(x, y)</code>	数値 <code>x</code> を <code>y</code> の桁数に整形し文字列型に変換	<code>format(1234.555555, 3)</code>
<code>date_format(date, format)</code>	日付 <code>date</code> を指定の形式に変換	<code>date_format(now(), '%Y-%m-%d')</code>
<code>time_format(date, format)</code>	時刻 <code>time</code> を指定の形式に変換	<code>time_format('100:00:00', '%H %k %h %I %l')</code>

日付 / 時刻関数

関数	説明	例
<code>adddate(day, interval ex unit)</code>	日付に指定した日数を加算	<code>adddate('2001-04-10', interval 10 day)</code>
<code>subdate(day, interval ex unit)</code>	日付に指定した日数を減算	<code>subdate('2001-04-10', interval 10 day)</code>
<code>current_date()</code>	本日の日付	
<code>current_time()</code>	本日の時刻	
<code>current_timestamp()</code>	日付と時刻	
<code>localtime()</code>	本日の時刻	
<code>localtimestamp()</code>	日付と時刻	
<code>now()</code>	現在の日付と時刻 (<code>current_timestamp</code> と同じ)	
<code>timediff(day1, day2)</code>	日付の差を算出	<code>timediff('2001-04-10', '2010-04-10')</code>

集約関数

関数	返り値型	説明
avg(expression)	integer 型引数であれば numeric、浮動小数点数の引数であれば double precision、それ以外は引数のデータ型と同じ	すべての入力値の平均値（算術平均）
count(*)	bigint	入力値の数値
count(expression)	bigint	expression の値が NULL でない入力値の個数
max(expression)	引数の型と同じ	すべての入力値間での expression の最大値
min(expression)	引数の型と同じ	すべての入力値間での expression の最小値
sum(expression)	smallint または integer 型の引数であれば bigint、bigint 型の引数であれば numeric、浮動小数点数の引数であれば double precision、それ以外は引数のデータ型と同じ	すべての入力値に渡って expression の和