

基本SQL構文と型：数値

- ▶ 数値は、さらに以下に分類されます
- ▶ 整数型：整数を扱います
 - ▶ TINYINT、SMALLINT(、MEDIUMINT)、INT、BIGINT
 - ▶ 及び、上述の符号なし(UNSIGNED)
- ▶ 小数：小数を扱います
 - ▶ 浮動小数点型：高速だが誤差が出る
 - ▶ DOUBLE(、FLOAT)
 - ▶ 固定小数点型：正確な精度だが処理が重い
 - ▶ DECIMAL(、NUMERIC)

基本SQL構文と型：数値：整数型

- ▶ TINYINT、SMALLINT(、MEDIUMINT)、INT、BIGINTは、以下の範囲の値を扱うことができます

型	ストレージ	最小値	最大値
TINYINT	1	-128	127
(UNSIGNED)	1	0	255
SMALLINT	2	-32768	32767
(UNSIGNED)	2	0	65535
MEDIUMINT	3	-8388608	8388607
(UNSIGNED)	3	0	16777215
INT	4	-2147483648	2147483647
(UNSIGNED)	4	0	4294967295
BIGINT	8	-9223372036854775808	9223372036854775807
(UNSIGNED)	8	0	18446744073709551615

基本SQL構文と型：数値：浮動小数点型

- ▶ 一般的にプログラムでよく用いられる「浮動小数点型」で数値を保持します
 - ▶ **DOUBLE**: 倍精度値(8バイト)、**FLOAT**: 単精度値(4バイト)
- ▶ 現在のマイクロプロセッサ(CPU)では「専用の回路」が組み込まれている事も多く、演算速度は概して高速です
- ▶ ただし浮動小数点数型は「(多くの値で)誤差がでる」ので、精度の高い情報を扱う際には注意が必要です
 - ▶ 誤差を減らすために、通常は**DOUBLE**が使われ、**FLOAT**が使われることは極端に減っているかと思います

基本SQL構文と型：数値：固定小数点型

- ▶ 固定小数点型は「誤差を出したくない小数点を含む値の扱い」に用いられます
 - ▶ 典型的には「金銭データ」でよく用いられます
- ▶ 浮動小数点数型と比較して「とても処理が重い」ので、使用には注意が必要です

基本SQL構文と型：文字

- ▶ 文字列型、という言い方をします
- ▶ 大まかに、以下があります
 - ▶ CHARおよびVARCHAR
 - ▶ BINARYおよびVARBINARY
 - ▶ TEXT
 - ▶ BLOB
- ▶ 文字列型は「文字コード」も重要になってきますので、意識しておきましょう
 - ▶ 昨今ですとMySQLでは「utf8mb4」という文字コードがもっとも基本になります
 - ▶ ただしutf8mb4が使えるのは「MySQL 5.5.3以降」なので注意

基本SQL構文と型：文字：CHAR/VARCHAR

- ▶ 文字列を扱う型としては最も基本的な型になります
- ▶ CHARは「長さ固定」となり、足りない分は右側にスペースが埋められます
- ▶ VARCHARは可変長の文字列になります
 - ▶ 昔はCHARがよく使われていましたが、最近はVARCHARを使う事のほうが多いかと思います
- ▶ MySQLとMariaDB、またバージョンによっても幾分差異がありますが、指定できる文字長には上限があります
- ▶ CHARおよびVARCHARの指定サイズは「文字長」です。たとえば「UTF-8であ」は「1文字」なので1となります

基本SQL構文と型：文字：

BINARY/VARBINARY

- ▶ MySQLに固有の型になり、CHAR/VARCHARと同様に「文字列を扱う型」になります
- ▶ BINARYは「長さ固定」となり、足りない分は右側にナル文字 (x00) が埋められます
- ▶ VARBINARYは可変長の文字列になります
- ▶ MySQLとMariaDB、またバージョンによっても幾分差異がありますが、指定できる文字長には上限があります
- ▶ BINARYおよびVAR BINARYの指定サイズは「バイト数」です。たとえば「UTF-8であ」は「3バイト」なので3となります
- ▶ 用途は「基本SQL構文と型：構文（次チャプター）」でやります

基本SQL構文と型：文字：TEXT/BLOB

- ▶ 「大きなサイズ」の文字列（またはバイナリデータ）を扱う時用の型になります
 - ▶ TEXTは多くのRDBにあり、BLOBはMySQL固有になります
- ▶ 型は以下の各4種類が指定できます
 - ▶ TINYTEXT、TEXT、MEDIUMTEXT、LONGTEXT
 - ▶ TINYBLOB、BLOB、MEDIUMBLOB、LOB
- ▶ LONG * は「4GB」までのサイズの文字列（バイナリデータ）を格納する事が出来る
- ▶ TEXT系はCHAR系に、BLOB系はBINARY系に近い特徴を持ちます（詳細は後程）

基本SQL構文と型：日付

- ▶ 日付型は、さらに以下に分かれます
 - ▶ 日付型：日付(年月日)を扱います
 - ▶ DATE
 - ▶ 時刻型：時刻(時分秒)を扱います
 - ▶ TIME
 - ▶ 日時型：日時(年月日時分秒)を扱います
 - ▶ DATETIME、TIMESTAMP

基本SQL構文と型：日付：日付と日時

▶ 日付型:DATE

- ▶ YYYY-MM-DDのフォーマットで扱われます
- ▶ MySQLは「'1000-01-01' から '9999-12-31'」の範囲が取り扱えます
 - ▶ MariaDBは、1000年よりも古い日付を扱えるようです

▶ 時刻型:TIME

- ▶ HH:MM:SSのフォーマットで扱われます
- ▶ 内部的には「'-838:59:59' から '838:59:59'」の範囲が取り扱えます
 - ▶ MySQL 5.6.4 以降は、マイクロ秒をサポートしています

-
- ▶ 日時型: DATETIME、TIMESTAMP
 - ▶ DATETIMEは「'1000-01-01 00:00:00' から '9999-12-31 23:59:59'」の範囲が扱えます
 - ▶ MariaDBは、1000年よりも古い日付を扱えるようです
 - ▶ TIMESTAMPは「'1970-01-01 00:00:01' UTC から '2038-01-19 03:14:07' UTC」の範囲が扱えます
 - ▶ いわゆる「2038年問題」があります
 - ▶ 上述から、MySQLでは日時を取り扱う時は原則として「DATETIME」のほうが適切です
 - ▶ 「時刻自動更新が欲しくてTIMESTAMP」という選択肢もありましたが、MySQL5.6.5以降、DATETIMEでもサポートされています