

A blue 3D cylinder representing a database, with the MySQL logo on its side. The logo consists of the word 'MySQL' in a stylized font, with 'My' in blue and 'SQL' in orange. The cylinder has a white highlight on its top edge and a shadow on the surface below it.

Data Types!

Ellie and Ethan

Numeric

Why do we need numeric data type?

Because we use numbers in everything! It is super important that we have a way to hold and transfer numerical information.

Exact numeric types - Integers

This includes “integer”, “tiny integer”, “small integer”, “medium integer”, and “big integer”. “tiny int” is used as a boolean where 0 = false and not 0 = true

```
die( "PHP 5.2 or greater is required!!!" );
```

Type	Storage (Bytes)	Minimum Value Signed	Minimum Value Unsigned	Maximum Value Signed	Maximum Value Unsigned
TINYINT	1	-128	0	127	255
SMALLINT	2	-32768	0	32767	65535
MEDIUMINT	3	-8388608	0	8388607	16777215
INT	4	-2147483648	0	2147483647	4294967295
BIGINT	8	-2 ⁶³	0	2 ⁶³ -1	2 ⁶⁴ -1

```
// Load configuration
```

‘Unsigned’ is a keyword that can be assigned to an integer to make it so that it doesn’t allow negative values

```
$tpl = new Template("/templates/html/error_config.html");  
echo $tpl->fetch();  
die();
```

Exact numeric types - Decimals/Numeric

82097494459230781640628620899862803482534211706798214

This includes “decimal”, and “numeric”, which act the same way

Store exact numeric data, used when it is important to be very precise

```
salary DECIMAL(5,2)
```

564823378678316527120190914564856692346

This says that there are 5 digits and 2 are behind the decimal point, making it a scale of -999.99 to 999.99. The max # of digits for any decimal is 65, and the max # of digits behind a decimal point is 30.

21861173819326117931051185480744623799627495673518857

52724891227938183011949129833673362440656643086021...

Approximate numeric types

This includes “float”, “real”, and “double precision”. “Real” and “double precision” are the same value.

Type	Bits	Have up to	Approximate Range
float	32	7 digits	$-3.4 \times 10^{(38)}$ to $+3.4 \times 10^{(38)}$
double	64	15-16 digits	$\pm 5.0 \times 10^{(-324)}$ to $\pm 1.7 \times 10^{(308)}$

Float has an optional precision value of float (p) where the value could be between 0-23 values. Also, a person could do float (m,d), real (m,d), or double precision (m,d); where m is the number of digits and d is the number of digits behind the decimal point.

Bit-Type Value

This includes the type “bit”, it stores bit values in binary form. For example, `b'111'` and `b'10000000'` represent 7 and 128, respectively.

To make a precision variable, you can do `bit(M)`, and M can range from 1 to 64 digits.

hg\$

hg\$ mysql

Wel

You

Server version: 5.0.12 MySQL Community Server - GPL

Cop Let's do some practice!

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> USE test;

Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed

mysql> SHOW tables;

+-----+	
Tables_in_test	
+-----+	



Strings

CHAR and VARCHAR

Declaring:

```
myString CHAR(45)          myString VARCHAR(45)
```

The 45 represents the maximum length of the string. The CHAR length can be from 0-255. The VARCHAR can be from 0-65535.

VARCHAR truncates spaces at the end. VARCHAR will only store bytes that are used, whereas CHAR will store all the bytes determined by the max length, even if they are zeros.

BINARY and VARBINARY

Declaring:

```
myString BINARY(10)      myString VARBINARY(10)
```

10 specifies the maximum number of bytes used to store the string. BINARY and VARBINARY are respectively similar to CHAR and VARCHAR besides this.

Binary system

0	1	0	1	0	1	0	1
<u>128</u>	<u>64</u>	<u>32</u>	<u>16</u>	<u>8</u>	<u>4</u>	<u>2</u>	<u>1</u>
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

BLOB and TEXT

BLOB is like VARBINARY and TEXT is like VARCHAR.

Both have 4 subtypes: TINY____, _____, MEDIUM____, and LONG_____.

The lengths of BLOB and TEXT are not restricted.



ENUM

Declaring:

```
myEnum ENUM('string 0', 'string 1', 'string2')
```

ENUM is like a list of strings.

You can get an element by saying:

```
WHERE myEnum='string1';
```

OR

```
WHERE myEnum=1;
```

SET

Declaring:

```
mySet SET('string 0', 'string 1', 'string 2')
```

SET is like ENUM, but ENUM only lets you select one element at a time. SET lets you select multiple elements in the set. It will return as **ONE** string, and the values will be concatenated and separated by a comma in the order they were declared in the table creation without duplicates.

SET does not allow multiples of the same string, and trailing whitespace is truncated.