Рассмотрим числовые типы данных.

• ***BIT***[(<Количествобитов>)].

Битовое число, содержащее заданное количество битов. Если количество битов не ука-

зано, число состоит из одного бита.

• ***TINYINT***.

Целое число в диапазоне либо от -128 до 127, либо (если указано свойство UNSIGNED)

от 0 до 255.

• ***BOOL*** или ***BOOLEAN***.

Являются синонимами к типу данных TINYINT(1) (число в скобках – это количество

отображаемых цифр, см. примечание ниже). При этом ненулевое значение рассматривается

как истинное (TRUE), нулевое – как ложное (FALSE).

• ***SMALLINT***.

Целое число в диапазоне либо от -32 768 до 32 767, либо (если указано свойство

UNSIGNED) от 0 до 65 535.

• ***MEDIUMINT***.

Целое число в диапазоне либо от -8 388 608 до 8 388 607, либо (если указано свойство

UNSIGNED) от 0 до 16 777 215.

• ***INT*** или ***INTEGER***.

Целое число в диапазоне либо от -2 147 483 648 до 2 147 483 647, либо (если указано

свойство UNSIGNED) от 0 до 4 294 967 295.

• ***BIGINT***.

Целое число в диапазоне либо от -9 223 372 036 854 775 808 до 9 223 372 036 854 775

807, либо (если указано свойство UNSIGNED) от 0 до 18 446 744 073 70 9 551 615.

• ***SERIAL***.

Синоним выражения BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT UNIQUE

(большое целое число без знака, принимающее автоматически увеличиваемые уникаль-

ные значения; значения NULL запрещены). Используется для автоматической генера-

ции уникальных значений в столбце первичного ключа. Описание свойств UNSIGNED и

AUTO\_INCREMENT вы найдете в этом подразделе, а свойств NOT NULL и UNIQUE – в

пункте «Свойства столбцов».

Для всех целочисленных типов данных, кроме BOOL (BOOLEAN) и

SERIAL, можно в скобках указать количество отображаемых цифр, которое

используется совместно с параметром ZEROFILL: если число содержит

меньшее количество цифр, то при выводе оно дополняется слева нулями.

Например, если столбец таблицы определен как INT (5) ZEROFILL, то

значения «1234567» и «12345» отображаются «как есть», а значение «123»

– как «00123». Для типа данных BIT в скобках указывается размер числа, то

есть максимальное количество хранимых битов.

• ***FLOAT***.

Число с плавающей точкой в диапазоне от -3,40282346638 до -1,175494351-38 и от

1,175494351-38 до 3,40282346638 (а также значение 0) с точностью около 7 значащих цифр

(точность зависит от возможностей вашего компьютера).

• ***DOUBLE***, ***DOUBLE PRECISION*** или ***REAL***.

Число с плавающей точкой в диапазоне от -1,7976931348623157308 до

-2,2250738585072014-308 и от 2,2250738585072014-308 до 1,797693134862315738 (а также зна-

чение 0) с точностью около 15 значащих цифр (точность зависит от возможностей вашего

компьютера).

• ***FLOAT***(<Точность>).

При значении точности от 0 до 24 этот тип данных эквивалентен типу FLOAT, при

значении от 25 до 53 – типу DOUBLE.

• ***DECIMAL***, ***DEC***, ***NUMERIC*** или ***FIXED***.

Точное (неокругляемое) число с фиксированной точкой. Может содержать до 65 знача-

щих цифр и до 30 цифр после десятичного разделителя (по умолчанию – 10 значащих цифр

и 0 после десятичного разделителя).

Для всех десятичных (нецелочисленных) типов данных, кроме

FLOAT(<Точность>), можно в скобках указать точность и шкалу, то

есть максимальное количество хранимых значащих цифр и максимальное

количество хранимых цифр после десятичного разделителя. Для чисел с плавающей точкой можно указать точность до 255

и шкалу до 30, однако указывать слишком большую точность и шкалу не

имеет смысла, так как в базе данных сохраняются приближенные значения,

которые совпадают с реальными лишь в первых 7 (для типа FLOAT) или 15

(для типа DOUBLE) значащих цифрах, последующие цифры при сохранении

могут быть искажены. Для чисел с фиксированной точкой можно указать

точность до 65 и шкалу до 30. Если точность и шкала не указаны, то они

равны, соответственно, 10 и 0. При сохранении чисел с фиксированной

точкой искажений не происходит.

Завершая рассмотрение числовых типов данных, обсудим три свойства, которые

можно указать для числовых столбцов:

• UNSIGNED – данное свойство означает, что в столбце запрещены отрицательные

(со знаком «-») значения. Указывать это свойство можно для любых столбцов с числовым

типом данных, кроме BIT, BOOL (BOOLEAN) и SERIAL. Для целочисленных столбцов при

добавлении свойства UNSIGNED максимально допустимое значение столбца увеличивается

вдвое.

• ZEROFILL – данное свойство означает, что значения при отображении будут допол-

нены нулями. Целые числа дополняются нулями слева в соответствии с указанным коли-

чеством отображаемых цифр, десятичные – слева и справа в соответствии с указанными

точностью и шкалой. Например, если столбец определен как DOUBLE(10,5) ZEROFILL, то

значение «12.23» отображается как «0012.23000». Кроме того, данное свойство запрещает

отрицательные значения, как и свойство UNSIGNED. Указывать свойство ZEROFILL можно

для любых столбцов с числовым типом данных, кроме BIT, BOOL (BOOLEAN) и SERIAL.

• AUTO\_INCREMENT – данное свойство обеспечивает автоматическую нумерацию

строк таблицы. Это означает, что при добавлении в столбец неопределенного (NULL) или

нулевого значения оно автоматически заменяется следующим номером, на единицу больше

предыдущего (нумерация по умолчанию начинается с единицы, установить другой началь-

ный номер можно с помощью соответствующего свойства таблицы). Указывать это свой-

ство можно для любых столбцов с числовым типом данных, кроме BIT и DECIMAL (DEC,

NUMERIC, FIXED). В таблице может быть только один столбец с таким свойством, и для

него должен быть создан ключ или индекс (об этом вы узнаете в пункте «Ключевые столбцы

и индексы»).

Для столбца, который будет содержать дату и/или время, вы можете использовать один

из следующих типов данных.

• ***DATE***.

Дата в формате «YYYY-MM-DD», в диапазоне от «0000-01-01» до «9999-12-31».

• ***DATETIME***.

Дата и время в формате «YYYY-MM-DD HH:MM: SS» в диапазоне от «0000–0101

00:00:00» до «9999-12-31 23:59:59».

• ***TIMESTAMP***.

Отметка времени в формате «YYYY-MM-DD HH:MM: SS» в диапазоне от «1970-01-01

00:00:00» до некоторой даты в 2038 г. При добавлении или изменении строки таблицы в

столбце с типом TIMESTAMP автоматически устанавливается дата и время выполнения опе-

рации (если значение этого столбца не указано явно или указано неопределенное значение).

Если нужно, чтобы отметка времени проставлялась только при добавлении строки, после

слова TIMESTAMP добавим свойство DEFAULT CURRENT\_TIMESTAMP.

Если в таблице есть несколько столбцов с типом TIMESTAMP, отметка времени авто-

матически проставляется только в первом из них. Если необходимо также вносить отметку

времени в какой-либо из последующих столбцов с типом TIMESTAMP, то при добавле-

нии/изменении строки укажем для этого столбца значение NULL, которое будет автомати-

чески заменено текущей датой.

• ***TIME***.

Время в формате «HH:MM: SS» в диапазоне от «-838:59:59» до «838:59:59».

• ***YEAR***, ***YEAR*** (2), ***YEAR*** (4).

Год в формате «YYYY» или «YY» (если количество цифр не указано, использу-

ется формат «YYYY»). Диапазон значений – от 1901 до 2155, если используется формат

«YYYY», или от 70 (соответствует 1970 г.) до 69 (соответствует 2069 г.), если используется

формат «YY».

Отмечу, что MySQL воспринимает даты не только в указанном выше формате.

Вы можете ввести дату с любым знаком препинания в качестве разделителя, например

2007@12@31 23%59%59, или без разделителя, например, 20071231235959. Более того, если

в столбец с типом даты или времени вносится символьное или числовое значение в одном

из таких форматов, MySQL автоматически преобразует это значение в дату и/или время.

Столбцам, которые будут содержать текст, можно присвоить один из следующих типов

данных.

• ***CHAR*** (<Количество символов>) или ***NATIONAL CHAR*** (<Количество символов>).

Символьная строка фиксированной длины. В таком столбце всегда хранится указан-

ное количество символов, при необходимости значение дополняется справа пробелами. Вы

можете задать количество символов от 0 до 255. Если количество символов не задано,

используется длина строки по умолчанию – 1 символ.

Тип данных NATIONAL CHAR отличается от CHAR тем, что для столбцов с типом

NATIONAL CHAR используется кодировка UTF-8, в то время как для столбцов с типом

CHAR можно указать любую кодировку, поддерживаемую MySQL.

• ***VARCHAR*** (<Максимальное количество символов>) или ***NATIONAL***

***VARCHAR*** (<Максимальное количество символов>).

Символьная строка переменной длины, содержащая не более указанного количества

символов. Вы можете указать максимальное количество символов от 0 до 65 535, но не более

65 535 байтов в сумме для всех столбцов таблицы с типом CHAR, VARCHAR, BINARY

или VARBINARY. Таким образом, если во всей таблице вы используете однобайтовую коди-

ровку (где каждому символу соответствует 1 байт, например кодировку KOI8-R, CP-866 или

CP-1251), то суммарное количество символов, указанное при описании этих столбцов, не

должно превышать 65 535. Если же вы используете кодировку UTF-8 (для которой сервер

MySQL выделяет до 3 байтов на символ), то суммарное количество символов, указанное при

описании этих столбцов, не должно превышать 21 844 (в три раза меньше, чем для однобай-

товых кодировок).

Тип данных NATIONAL VARCHAR отличается от VARCHAR тем, что для столбцов с

типом NATIONAL VARCHAR используется кодировка UTF-8, в то время как для столбцов

с типом VARCHAR можно указать любую кодировку, поддерживаемую MySQL.

• ***BINARY*** (<Количество байтов>)

Байтовая (бинарная) строка фиксированной длины. Этот тип аналогичен типу CHAR,

только строка содержит не символы, а байты, и значение меньшей длины дополняется справа

не пробелами, а нулевыми байтами.

• ***VARBINARY*** (<Максимальное количество байтов>)

Байтовая (бинарная) строка переменной длины. Этот тип аналогичен типу VARCHAR,

только строка содержит не символы, а байты.

• ***TINYBLOB***

Байтовая (бинарная) строка переменной длины. Максимальная длина – 255 байтов.

• ***TINYTEXT***

Символьная строка переменной длины. Максимальная длина – 255 байтов (не симво-

лов!).

**Примечание**

Обратите внимание, что для типов данных TINYTEXT, TEXT,

MEDIUMTEXT или LONGTEXT длина значения ограничена максимальным

количеством байтов, а не символов. Для однобайтовых кодировок (таких

как KOI8-R, CP-866 или CP-1251) длина значения в байтах и в символах

одинакова. Однако для многобайтовых кодировок реальное количество

символов в значении может быть меньше, чем количество байтов. Так,

в кодировке UTF-8 для кодирования символов английского алфавита

используется 1 байт на символ, для русского алфавита – 2 байта на символ,

поэтому максимальное количество символов русского алфавита, которое

можно ввести в такой столбец, приблизительно в два раза меньше, чем

максимальное допустимое количество байтов для этого столбца.

• ***BLOB*** [(<Максимальное количество байтов>)].

Байтовая (бинарная) строка переменной длины. Если количество байтов не указано, то

значение столбца ограничено 65 535 байтами. Если количество байтов указано, то создается

столбец с типом данных TINYBLOB, BLOB, MEDIUMBLOB или LONGBLOB: выбирается

тип данных с наименьшим размером, достаточным для хранения этого количества байтов.

• ***TEXT*** [(<Максимальное количество символов>)].

Символьная строка переменной длины. Если количество символов не указано, то зна-

чение столбца ограничено 65 535 байтами. Если количество символов указано, то создается

столбец с типом данных TINYTEXT, TEXT, MEDIUMTEXT или LONGTEXT: выбирается

тип данных с наименьшим размером, достаточным для хранения этого количества символов.

• ***MEDIUMBLOB***.

Байтовая (бинарная) строка переменной длины. Максимальная длина – 16 777 215 бай-

тов.

• ***MEDIUMTEXT***.

Символьная строка переменной длины. Максимальная длина – 16 777 215 байтов.

• ***LONGBLOB***.

Байтовая (бинарная) строка переменной длины. Максимальная длина – не более 4

294 967 295 байтов (4 Гбайт), в зависимости от используемого протокола взаимодействия с

сервером MySQL и доступных системных ресурсов.

• ***LONGTEXT***.

Символьная строка переменной длины. Максимальная длина – не более 4 294 967 295

байтов (4 Гбайт), в зависимости от используемого протокола взаимодействия с сервером

MySQL и доступных системных ресурсов.

• ***ENUM*** ('<Значение 1>', '<Значение 2>' ...).

Строка, содержащая ровно один элемент из заданного списка. Например, если стол-

бец определен как ENUM('a','b'), то допустимыми значениями этого столбца являются зна-

чения a, b и NULL (а также пустая строка «», которая может появиться при попытке вставки

некорректного значения в данный столбец; о добавлении строк в таблицу и о возможных

вариантах обработки некорректных значений пойдет речь в подразделе «Вставка отдельных

строк»). В список вы можете включить до 65 535 элементов.

• ***SET*** ('<Значение 1>', '<Значение 2>' ...).

Строка, содержащая любой набор элементов из заданного списка (в том числе

пустой). Например, если столбец определен как SET('a','b'), то он может содержать значения

«» (пустая строка), a, b, a, b и NULL. В список вы можете включить до 64 элементов. Эле-

менты списка не должны содержать запятых. Каждый из элементов может присутствовать в

значении столбца только один раз, причем элементы могут следовать только в том порядке,

в котором они перечислены в списке. Например, при вставке значений a, b, a, b и b, a они авто-

матически преобразуются в значение a, b.

В заключение отметим, что в MySQL вы можете указать кодировку отдельно для

каждого символьного столбца. А именно, для столбцов с типом CHAR, VARCHAR,

TINYTEXT, TEXT, MEDIUMTEXT, LONGTEXT, ENUM и SET вы можете задать свойство

CHARACTER SET <Имя кодировки> и/или COLLATE <Имя правила сравнения> (подроб-

нее о кодировках и правилах сравнения символьных значений говорилось в разделе «Созда-

ние базы данных»).

Например, чтобы имена клиентов хранились в кодировке CP-1251, тогда как кодиров-

кой по умолчанию для таблицы Customers (Клиенты) является UTF-8, столбец name (имя)

можно определить следующим образом:

name ***VARCHAR*** (100) ***CHARACTER*** ***SET*** cp1251 COLLATE cp1251\_general\_ci;