

# MySQL i typy danych

Przystępując do projektowania tabel bazy danych, konieczne jest określenie typu danych dla każdej z kolumn tabeli. MySQL podobnie jak inne systemy zarządzania bazą danych opiera się na trzech kategoriach danych:

- danych typu znakowego (tekstowego);
- liczbowego;
- daty i czasu.

W każdej z wymienionych grup wyróżnia się kilka odmian typów danych, z których pewne są charakterystyczne jedynie dla MySQL. Właściwy wybór typu dla danej kolumny tabeli wpływa nie tylko na rodzaj danych, jakie mogą być w niej gromadzone ale także na sposób ich przechowywania, oraz na całkowitą wydajność bazy danych.

*Należy pamiętać, że wielkość pamięci potrzebnej do przechowania jednej danej ściśle zależy od jej typu. Deklarując typ kolumny, należy kierować się następującą zasadą: zawsze wybiera się taki typ danych, który będzie zajmował najmniej pamięci i jednocześnie pozwoli na zapisanie wszystkich potrzebnych danych.*

Podstawowe typy danych MySQL przedstawiono w poniższych tabelach:

## Dane liczbowe

Typy liczbowe dzielą się na całkowitoliczbowe i zmiennoprzecinkowe. W przypadku tych ostatnich istnieje możliwość zadeklarowania liczby cyfr znaczących po przecinku.

Nazwa typu danych	Opis typu danych
<b>TINYINT</b>	1-bajtowe pole, przechowujące wartości całkowite bez znaku z przedziału od 0 do 255 lub ze znakiem z przedziału od -127 do 127
<b>SMALLINT</b>	2-bajtowa wartość całkowita. Zakres wartości bez znaku od 0 do 65 535 lub ze znakiem od -32 768 do 32 768
<b>MEDIUMINT</b>	3-bajtowa wartość całkowita. Zakres wartości od 0 do 16 777 215 lub ze znakiem od -8 388 608 do 8 388 608
<b>INT</b>	4-bajtowa wartość całkowita. Zakres wartości ze znakiem od -2 147 483 648 do 2 147 483 647 lub bez znaku od 0 do 4 294 967 295
<b>BIGINT</b>	8-bajtowa wartość całkowita. Typ BIGINT jest używany podczas przeprowadzania obliczeń. Stosując pola tego typu we własnej bazie danych należy uważać, aby ich wartości nie były zbyt duże, ponieważ użyte w obliczeniach mogą doprowadzić do błędów przepełnienia

<i>Nazwa typu danych</i>	<i>Opis typu danych</i>
<b>FLOAT</b>	4-bajtowa liczba zmiennoprzecinkowa z zakresu od -1.402823466E+38 do -1.175494351E-38, 0 i 1.175494351E-38 do 3.402823466E+38
<b>DOUBLE</b>	8-bajtowa liczba zmiennoprzecinkowa przechowująca wartości z zakresu od -1.7976931348623157E+308 do -2.2250738585072014E-308, 0, i 2.2250738585072014E-308 do 1.7976931348623157E+308
<b>DECIMAL</b>	Liczba zmiennoprzecinkowa o zakresie typu DOUBLE. Liczba przechowywana jako łańcuch.

Typy liczbowe mogą być dodatkowo opisane atrybutem **UNSIGNED** co ogranicza wartości kolumny do liczb dodatnich i zera. Oznacza to, że do kolumny będzie można wprowadzać wyłącznie wartości dodatnie lub zero. Kolumny typu liczbowego mogą także otrzymać atrybut **ZEROFILL**, co oznacza, że przy wyświetlaniu zawartości kolumn posiadających ten atrybut, zapisane w nich liczby zostaną poprzedzone zerami (np. jeśli długość danej w polu określono na **4** to po wpisaniu liczby **15** zostanie ona wyświetlona jako **0015**). Jeżeli kolumna zostanie zdefiniowana jako **ZEROFILL**, automatycznie będzie także ona również miała uaktywniony atrybut **UNSIGNED**.

## Dane daty i czasu

MySQL obsługuje szereg typów daty i czasu. Każdy z nich pozwala na wpisanie danych w formie liczbowej lub łańcucha znaków.

<i>Nazwa typu danych</i>	<i>Opis typu danych</i>
<b>DATE</b>	Data (bez czasu), wyświetlana w formacie RRRR-MM-DD. Akceptuje daty w różnych formatach z zakresu od 1000-01-01 do 9999-12-31
<b>DATETIME</b>	Pole daty z czasem dnia wyświetlane według formatu RRRR-MM-DD GG:MM:SS. Może przechowywać wartości z zakresu od 1000-01-01 00:00:00 do 9999-12-31 23:59:59
<b>TIMESTAMP</b>	Data i czas liczony od początku epoki systemu UNIX, 1970-01-01 00:00:00, do momentu kiedy 32-bitowe pole przechowujące liczbę sekund, jaka upłynęła od tej daty ulegnie przepełnieniu w roku 2037. Jeśli polu temu nie została nadana wartość przez wyrażenie INSERT lub UPDATE, domyślnie otrzyma ono wartość daty bieżącej
<b>TIME</b>	Czas mieszczący się w przedziale od -838:59:59 do 838:59:59
<b>YEAR</b>	Rok. Dozwolone wartości to przedział od 1901 do 2155 oraz wartość 0000. MySQL wyświetla wartość typu YEAR w formacie RRRR

## Dane znakowe

Typy łańcuchowe dzielą się na trzy grupy: klasyczne łańcuchy, łańcuchy typu **TEXT** i **BLOB** oraz łańcuchy typu **ENUM** i **SET**.

### 1. Typy danych klasycznych łańcuchów znaków

Klasyczne (zwykłe) łańcuchy znaków oznaczają krótkie fragmenty tekstu (frazy, np. imiona, nazwiska, nazwy, kody pocztowe itd.). Do tej grupy zaliczamy typy **CHAR** (łańcuchy o stałej długości) oraz **VARCHAR** (łańcuchy o zmiennej długości). Dla każdego z nich można określić długość łańcucha.

Nazwa typu danych	Opis typu danych
<b>CHAR</b>	Pole znakowe o stałej długości z zakresu od 1 do 255 bajtów. Po wstawieniu wartości puste miejsca pola CHAR są uzupełniane z prawej strony spacjami
<b>VARCHAR</b>	Pole znakowe o zmiennej długości z zakresu od 1 do 255 bajtów. Zajmowany jest jedynie taki obszar pamięci, jakiego wymaga wartość wstawiona w to pole

Łańcuchy zapisywane w kolumnach typu **CHAR** zostają uzupełnione spacjami w celu pełnego wykorzystania dopuszczalnego limitu ich długości. Natomiast w kolumnach typu **VARCHAR** łańcuchy są zapisywane w ich pierwotnej formie, dzięki czemu zajmują one tylko tyle pamięci, ile potrzeba. W przypadku wartości **CHAR** MySQL usuwa końcowe znaki spacji przy ich pobieraniu z bazy, natomiast ewentualne znaki spacji znajdujące się na końcu łańcucha **VARCHAR** zostaną wyeliminowane w momencie zapisywania go w bazie. Wybór któregoś z tych typów sprowadza się więc do rozstrzygnięcia dylematu między zwiększeniem szybkości działania kosztem używanej pamięci a ograniczeniem zużycia pamięci i spadkiem wydajności systemu.

### 2. Typy danych długich łańcuchów znaków i danych binarnych

Do tej grupy należą typy **TEXT** i **BLOB**. Wartości obu wymagają zróżnicowanych ilości pamięci. Typy te są przeznaczone do przechowywania, odpowiednio, dłuższych tekstów (do 65535 znaków) oraz danych binarnych. Wartości typu **BLOB** to tzw. duże obiekty binarne, stąd też wywodzi się jego nazwa. Obiekty te mogą przechowywać każdy rodzaj danych, np. obrazy, dźwięki itp. W praktyce jedyna różnica pomiędzy oboma typami polega na tym, że w danych typu **BLOB** rozróżniana jest wielkość liter, typ **TEXT** natomiast tej właściwości nie posiada.

<i>Nazwa typu danych</i>	<i>Opis typu danych</i>
<b>TINYBLOB, TINYTEXT</b>	Kolumna binarna lub tekstowa o rozmiarze nie przekraczającym 255 bajtów
<b>BLOB, TEXT</b>	Kolumna binarna lub tekstowa o rozmiarze nie przekraczającym 65 535 bajtów
<b>MEDIUMBLOB, MEDIUMTEXT</b>	Kolumna binarna lub tekstowa o rozmiarze nie przekraczającym 16 777 215 bajtów
<b>LOB, LONGTEXT</b>	Kolumna binarna lub tekstowa o rozmiarze nie przekraczającym 4 294 967 295 bajtów

### 3. Typy ENUM i SET

Trzecia grupa składa się z dwóch typów specjalnych **SET** i **ENUM**. Typ **SET** jest używany w celu zawężenia wartości danych, które mogą być zapisane w danej kolumnie, do z góry określonego zbioru wartości, przy czym dane zapisane w kolumnie mogą mieć więcej niż jedną wartość z tego zbioru. Podany zbiór wartości może zawierać do 64 elementów. Typ **ENUM** jest podobny do typu **SET**, jednak kolumny typu **ENUM** mogą zawierać tylko jedną spośród określonego zbioru wartości lub wartość **NULL**, a zbiór wartości dopuszczalnych może zawierać do 65 535 elementów.

<i>Nazwa typu danych</i>	<i>Opis typu danych</i>
<b>ENUM('elem1', 'elem2', ...)</b>	Kolumna łańcuchowa, która może zawierać jedną z wartości wymienionych na liście wartości w definicji tej kolumny
<b>SET('elem1', 'elem2', ...)</b>	Kolumna łańcuchowa, która może zawierać dowolną liczbę wartości spośród wartości występujących na liście w definicji tej kolumny