

Banco de dados



TIPOS DE DADOS

Vamos entender os tipos de dados, que no caso se refere aos campos, ou seja, a colunas de uma tabela, então começaremos para o tipo inteiro, com segue abaixo.

TIPO	BYTES	MENOR VALOR	UNSIGNED	MAIOR VALOR	UNSIGNED
TINYINT	1	-128	0	127	255
SMALLINT	2	-32768	0	32767	65535
MEDIUMINT	3	-8388608	0	83888607	16777215
INT	4	-2147483648	0	2147483647	4294967295
BIGINT	8	-2xE63	0	2xE63-1	2xE64

Na tabela acima temos os tipos **inteiros**, que basicamente faz com que a coluna armazene esse tipo de dados, assim sendo, TINYINT é o menor para armazenamento, enquanto BIGINT é o maior para armazenamento. Vale entender que quando não têm sinal (Unsigned) significa dizer que somente haverá valores positivos, assim sendo, os tipos aumentam a quantidade de armazenamento dos inteiros.

O Ponto flutuante tem os tipos FLOAT e DOUBLE, onde faz com que o campo receba um valor quebrado ou podem ser de pontos fixos.

TIPO	VALOR
FLOAT	4 BYTES
DOUBLE	8 BYTES

O Ponto fixo é necessário especificar o número de dígitos e o número de casas decimais. Exemplo: Se declaramos um campo DECIMAL(5,2) iremos poder armazenar valores somente entre -999,99 e 999,99.

TIPO	DÍGITOS
DECIMAL	65 DÍGITOS
NUMERIC	

Temos o tipo BIT é um tipo de dados usado para armazenar valores de bit. O tamanho de um campo BIT pode variar de 1 a 64 bits. Por exemplo, se você declarar um campo como BIT(1), ele poderá armazenar apenas um valor de bit, que pode ser 0 ou 1. Se você declarar um campo como BIT(2), ele poderá armazenar dois valores de bit, que podem ser 00, 01, 10 ou 11.

TIPO	VALOR
BIT	1 ou 0

O tipo Data e Hora são usados quando você precisa armazenar informações de data e hora em um banco de dados. Por exemplo, você pode usar o tipo de dados DATE para armazenar a data de nascimento de um cliente ou o tipo de dados DATETIME para armazenar a data e hora em que um pedido foi feito.

TIPO	VALOR
DATE	1000-01-01 até 9999-12-31.
DATETIME	1000-01-01 00:00:00 até 9999-12-31 23:59:59
TIMESTAMP	1970-01-01 00:00:01 UTC até 2038-01-19 UTC
TIME	-838:59:59 e 839:59:59
YEAR	1901 – 2155

O tipo STRING é uma sequência de caracteres usada para representar texto. CHAR e VARCHAR são dois tipos de dados usados para armazenar valores de string em bancos de dados relacionais, como o MySQL. A principal diferença entre eles é a forma como eles armazenam e recuperam os dados.

TIPO	BYTES
CHAR	0~255
VARCHAR	0~255

CHAR: É usado para armazenar valores de string de tamanho fixo. Quando você define um campo CHAR, você especifica o tamanho máximo em bytes que ele pode armazenar. Por exemplo, se você declarar um campo como CHAR(10), ele poderá armazenar até 10 bytes de dados. Se o valor armazenado tiver menos de 10 bytes, o campo será preenchido com espaços em branco até atingir o tamanho máximo especificado

VARCHAR: É usado para armazenar valores de string de tamanho variável. Quando você define um campo VARCHAR, você especifica o tamanho máximo em bytes que ele pode armazenar. Por exemplo, se você declarar um campo como VARCHAR(10), ele poderá armazenar até 10 bytes de dados. No entanto, ao contrário do CHAR, o VARCHAR não preenche o campo com espaços em branco se o valor armazenado tiver menos bytes do que o tamanho máximo especificado.

No MySQL, existem quatro tipos de BLOB e quatro tipos de TEXT, que diferem apenas no comprimento máximo dos valores que podem armazenar¹. Aqui estão os detalhes sobre os comprimentos máximos e os requisitos de armazenamento:

- **TINYBLOB:** Um objeto grande binário com um comprimento máximo de 255 ($2^8 - 1$) bytes.
- **BLOB:** Um objeto grande binário com um comprimento máximo de 65535 ($2^{16} - 1$) bytes, ou 64KB em armazenamento².
- **MEDIUMBLOB:** Um objeto grande binário com um comprimento máximo de 16777215 ($2^{24} - 1$) bytes, ou 16MB em armazenamento².
- **LOB:** Um objeto grande binário com um comprimento máximo de 4294967295 ($2^{32} - 1$) bytes, ou 4GB em armazenamento².

- **TINYTEXT**: Um objeto grande de texto com um comprimento máximo de 255 ($2^8 - 1$) caracteres¹.
- **TEXT**: Um objeto grande de texto com um comprimento máximo de 65535 ($2^{16} - 1$) caracteres¹.
- **MEDIUMTEXT**: Um objeto grande de texto com um comprimento máximo de 16777215 ($2^{24} - 1$) caracteres¹.
- **LONGTEXT**: Um objeto grande de texto com um comprimento máximo de 4294967295 ($2^{32} - 1$) caracteres¹.

Os valores **BLOB** são tratados como strings binárias (strings de bytes). Eles têm o conjunto de caracteres binários e a ordenação e comparação são baseadas nos valores numéricos dos bytes nos valores da coluna.

Os valores **TEXT** são tratados como strings não-binárias (strings de caracteres). Eles têm um conjunto de caracteres diferente do binário e os valores são ordenados e comparados com base na ordenação do conjunto de caracteres¹.

Você deve usar os tipos BLOB quando precisar armazenar dados binários, como imagens, áudio ou vídeo. Por outro lado, você deve usar os tipos TEXT quando precisar armazenar grandes quantidades de texto, como descrições longas ou conteúdo de postagens em blogs.

Existem mais tipos, mas os que foram apresentados são bem utilizados.

Abaixo temos uma dados de cadastro de clientes, leia com a atenção e pense como poderia dar tipos a cada campo. Segue abaixo:

Cadastro de Clientes

- Identificação única
- CPF do cliente
- O nome completo
- Endereço (Rua, bairro, cidade, estado e CEP)
- Data de nascimento
- O sexo
- O Salário
- Se está ativo
- Data de Entrada
- Data de Saída

Poderíamos fazer dessa forma cada acima:

- Identificação única (**INT**)
- CPF do cliente (**VARCHAR(11)**)
- O nome completo (**VARCHAR(30)**)
- Endereço (Rua, bairro, cidade, estado e CEP) **TEXT**
- Data de nascimento (**DATE**)
- O sexo (**VARCHAR(20)**)
- O Salário (**FLOAT**)
- Se está ativo (**BIT**)

- Data de Entrada (**DATETIME**)
- Data de Saída (**DATETIME**)

Claro que pode ser melhorado acima, mas de forma bem genérica podemos utilizar esses tipos de dados caso não haja nenhuma técnica de abstração.

