**SENAI Roberto Mange – Desenvolvimento de Sistemas**

Prof. Marcia Scanacapra

**RELATÓRIO DE PESQUISA**

**Tipos de Dados no MySQL**

**Ícone

O conteúdo gerado por IA pode estar incorreto.**

**Layslla Eduarda Oreti dos Santos**

**Agosto/2025**

Sumário

[Tipos de dados 3](#_Toc207111846)

**[Numéricos](#_Toc207111847)** [3](#_Toc207111847)

[**Caractere** 4](#_Toc207111848)

[**Texto Completo** 5](#_Toc207111849)

[**Data e Hora** 5](#_Toc207111850)

[**Binários** 6](#_Toc207111851)

[**Intervalo** 7](#_Toc207111852)

[Referências bibliográficas: 8](#_Toc207111853)

**Tipos de Dados no MySQL**

O MySQL é um dos sistemas de gerenciamento de banco de dados relacionais mais conhecido, oferecendo uma variedade de tipos de dados para que os desenvolvedores possam armazenar os dados de vários tipos e formatos. A gestão de dados é extremamente importante para qualquer sistema, e um dos seus aspectos mais importante da gestão de dados é a escolha do tipo de dado adequado para cada valor que for armazenado na base de dados.

# Tipos de dados

# **Numéricos**

Os tipos de dados numéricos são usados para armazenar valores numéricos com diferentes tamanhos.

1. **INT** – É utilizado para armazenar valores que sejam inteiros.

INSERT INTO meus\_valores (valor\_inteiro) VALUES (25);

1. **BIGINIT** – É semelhante ao tipo INT, mas ele pode armazenar valores inteiros maiores, sendo adequado e usado para armazenar valores inteiros grandes, utilizado em tabelas com muitas linhas e colunas.

INSERT INTO meus\_valores (valor\_grande) VALUES (9223372036854775808);

1. **SMALLINT** – Utilizado para armazenar valores inteiros menores e é usado em tabelas com valores que são limitados.

INSERT INTO meus\_valores (valor\_pequeno) VALUES (-32768);

1. TINYINT: Ele é semelhante ao SMALLINT, porém ele pode armazenar valores inteiros em um intervalo de -128 a 127.

INSERT INTO meus\_valores (valor\_muito\_pequeno) VALUES (-128);

1. **DECIMAL** – Usado para armazenar valores numéricos com uma precisão que tenha sido especificada, sendo adequado para armazenar valores monetários e financeiros.

INSERT INTO meus\_valores (valor\_decimal) VALUES (123.45);

1. **FLOAT** – Armazena valores numéricos com uma precisão específica, ou seja, ele armazena números decimais de até 6 dígitos.

INSERT INTO meus\_valores (valor\_float) VALUES (3.1415926535897932385);

1. **DOUBLE** – Semelhante ao tipo FLOAT, mas pode armazenar valores numéricos com uma maior precisão, até 16 dígitos. Desse jeito, ele armazena valores decimais, como valores financeiros ou científicos.

INSERT INTO meus\_valores (valor\_double) VALUES (3.14159265358979323846);

# **Caractere**

Os tipos de dados de caractere são usados para armazenar dados de texto e caracteres.

1. **CHAR** – Usado para armazenar dados de textos fixos com um comprimento específico. Ele é usado ao criar tabelas que armazenam valores de texto com o mesmo tamanho em todas as colunas.

CREATE TABLE meus\_valores ( nome CHAR(15) );

1. **VARCHAR** – Ele é semelhante ao tipo CHAR, mas permite que os valores de texto sejam variáveis, com comprimentos diferentes em cada coluna. Ou seja, ele é usado para criar tabelas para armazenar valores de texto com comprimentos variáveis.

CREATE TABLE meus\_valores ( email VARCHAR(200) );

1. **TEXT** – Utiliza-se de dados do tipo TEXT para armazenar dados de texto longos e complexos, com comprimentos que variam de 1 a 4 GB. São usados para armazenar valores de texto com muito conteúdo, como documentos e artigos de blog.

INSERT INTO meus\_valores (descricao) VALUES ('Este é um exemplo de valor de texto longo');

1. **BLOB** – É utilizado para armazenar dados binários (imagens, vídeos e arquivos de documentos). Ele armazena valores de grande porte e pode armazenar até 65.535 bytes.

INSERT INTO meus\_valores (imagem) VALUES (FILE('/caminho/para/imagem.jpg'));

# **Texto Completo**

1. **FULLTEXT**: Usado para armazenar dados de texto e realizar consultas de texto completo. Ele suporta consultas de texto completo com vários campos e permite o uso de operações de colisão e proximidade para encontrar resultados que sejam mais precisos.

CREATE TABLE livros (

id INT PRIMARY KEY,

titulo TEXT,

autor TEXT,

conteudo TEXT,

KEY(conteudo)(FULLTEXT)

);

INSERT INTO livros (id, titulo, autor, conteudo) VALUES (1, 'Livro 1', 'Autor 1', 'Conteúdo 1, Conteúdo 2, Conteúdo 3');

# **Data e Hora**

Os tipos de dados data e hora são usados para armazenar valores de data e hora.

1. **DATE** – Usados para armazenar apenas a data sem a hora.

INSERT INTO meus\_valores (data) VALUES ('2022-03-14');

1. **TIME** – Utilizado para armazenar apenas a hora sem a data.

INSERT INTO meus\_valores (hora) VALUES ('13:30:00');

1. **DATETIME** – Utiliza-se o tipo de dados DATETIME para armazenar data e hora juntos.

INSERT INTO meus\_valores (data\_hora) VALUES ('2022-03-14 13:30:00');

1. **TIMESTAMP** – Ele é semelhante ao tipo DATETIME, mas inclui a precisão do segundo.

INSERT INTO meus\_valores (data\_hora\_segundos) VALUES ('2022-03-14 13:30:00');

# **Binários**

O MySQL suporta vários tipos de dados binários, cada um com as suas próprias características e utilidades.

1. **BIT** – É um tipo binário simples que pode conter um único valor de 0 ou 1.

CREATE TABLE pessoas (

id INT PRIMARY KEY,

sexo BIT);

INSERT INTO pessoas (id, sexo) VALUES (1, 1);

1. **BITMAP** – Ele é um tipo de dado binário mais complexo que permite armazenar vários valores de bits em uma única coluna.

CREATE TABLE pessoas (

id INT PRIMARY KEY,

hobbies BITMAP);

INSERT INTO pessoas (id, hobbies) VALUES (1, 0b00100000);

1. **SET** – O tipo SET permite que vários valores de bit sejam armazenados em uma única coluna. Ele é semelhante ao BITMAP, porém mais fácil de usar e oferece maior desempenho.

CREATE TABLE pessoas (

id INT PRIMARY KEY,

hobbies SET);

INSERT INTO pessoas (id, hobbies) VALUES (1, 'leitura, cinema');

1. **ENUM** – É um tipo binário que permite armazenar um valor de uma lista pré-definida de valores.

CREATE TABLE cores (

id INT PRIMARY KEY,

cor ENUM('verde', 'azul', 'roxo'));

INSERT INTO cores (id, cor) VALUES (1, 'verde');

# **Intervalo**

1. **YEAR** – É utilizado para armazenar um número de ano, armazena valores de 1 a 9999. Ele é muito usado para armazenar datas de nascimento ou a data de início de um contrato.

CREATE TABLE pessoas (

id INT PRIMARY KEY,

nascimento YEAR);

INSERT INTO pessoas (id, nascimento) VALUES (1, 1990);

1. **MONTH** – O tipo de dado MONTH é usado para armazenar um número de mês, ou seja, ele armazena valores de 1 a 12.

CREATE TABLE pessoas (

id INT PRIMARY KEY,

nascimento MONTH);

INSERT INTO pessoas (id, nascimento) VALUES (1, 1);

1. **DAY** – É usado para armazenar um número de dia da semana, nesse sentido, ele armazena valores de 0 a 6, no qual o 0 representa o domingo e 6 representa sábado.

CREATE TABLE pessoas (

id INT PRIMARY KEY,

nascimento DAY);

INSERT INTO pessoas (id, nascimento) VALUES (1, 0);

# Referências bibliográficas:

HOMEHOST. *Quais são os tipos de dados do MySQL ?*

Disponível em: https://www.homehost.com.br/blog/tutoriais/mysql/tipos-de-dados-do-mysql/

MySQL. *Chapter 13 Data Types*:

Disponível em:  <https://dev.mysql.com/doc/refman/8.4/en/data-types.html>

DIO. *Chapter Domine os tipos de dados do MySQL*:

Disponível em:  <https://www.dio.me/articles/domine-os-tipos-de-dados-do-mysql>