

A high-angle, top-down photograph of a man with dark hair, wearing a light blue button-down shirt, sitting at a desk. He is looking down at a computer keyboard. His right hand is on the keyboard, and his left hand is on a computer mouse. There are several computer monitors on the desk. One monitor in the foreground shows a colorful bar chart. Another monitor in the background shows a line graph. The desk is light-colored, and the floor is made of light wood. The overall lighting is bright and even.

Entendendo **DATATYPE**

Os tipos de dados no MySQL



hcode

MySQL 5.7

Data Type

Categorias

Numéricos | Data e Hora | Texto

Os tipos de Dados definem o formato das informações que deverão ser preenchidas nas colunas das tabelas.

Os tipos de dados são utilizados durante a *criação da tabela*.



Exemplo:

```
create table tb_funcionario (  
    id_func          int,  
    nome_func        varchar(200),  
    email_func        varchar(250),  
    vlr_sal_func      decimal(10,2)  
);
```



Exemplo:

```
create table tb_funcionario (  
    id_func int,  
    nome_func varchar(200),  
    email_func varchar(250),  
    vlr_sal_func decimal(10,2)  
);
```

Nome de Colunas

Não podem começar com números ou caracteres especiais, o único caractere especial permitido é _ "underline".

Para os nomes de colunas, é possível conter números no meio ou no fim.



Exemplo:

```
create table tb_funcionario (  
    id_func          int,  
    nome_func        varchar(200),  
    email_func        varchar(250),  
    vlr_sal_func      decimal(10,2)  
);
```



Exemplo:

```
create table tb_funcionario (  
    id_func int,  
    nome_func varchar(200),  
    email_func varchar(250),  
    vlr_sal_func decimal(10,2)  
);
```

Tipo de Dados

Os Datatypes garantem a integridade dos dados, permitindo que somente informações em um formato específico sejam inseridas. Por exemplo um campo salário que aceite caracteres alfanuméricos, poderia gerar o que é chamado de anomalia de dados.

Se em vez de alfanumérico for utilizado o tipo decimal, impedirá que informações fora desse padrão numérico seja utilizado.



Conheça os tipos de dados

NUMÉRICOS

Tipos Inteiros

❖ Numéricos Inteiros

tinyint

smallint

mediumint

int

bigint

❖ Numéricos Inteiros

tinyint 1 Byte
smallint 2 Byte
mediumint 3 Byte
int 4 Byte
bigint 8 Byte



Os tipos numéricos são
divididos entre

signed

Unsigned



Os tipos numéricos são
divididos entre

Este é padrão

signed

Unsigned



signed

NUMÉRICOS

Permite números negativos

unsigned

NUMÉRICOS

Não permite números negativos



❖ Numéricos Inteiros

tinyint -128

smallint -32768

mediumint -8388608

int -2147483648

bigint -9223372036854775808



❖ Numéricos Inteiros

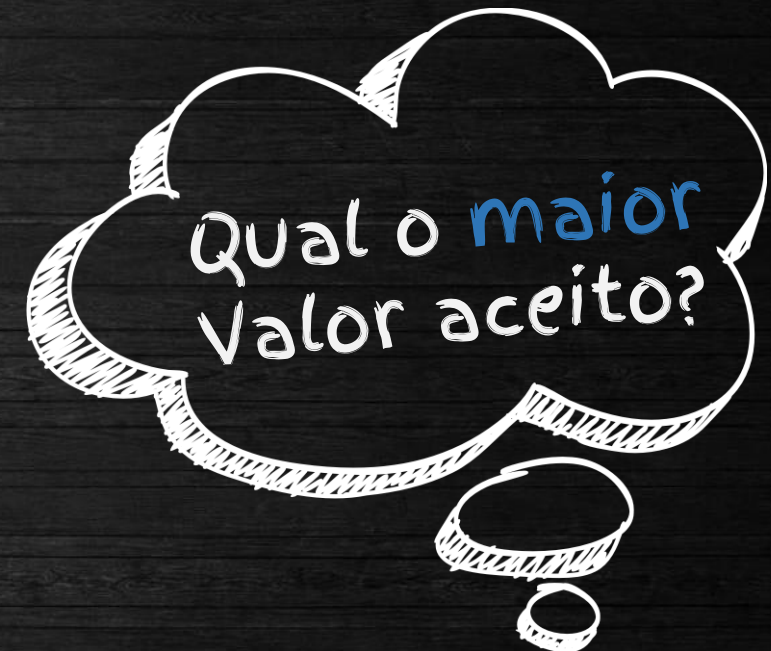
tinyint 127

smallint 32767

mediumint 8388607

int 2147483647

bigint 9223372036854775807



signed

NUMÉRICOS

O que significa os maiores e menores valores aceitos?

signed

NUMÉRICOS

Quais valores serão **permitidos** quando o usuário executar um comando **insert** e tentar cadastrar uma informação dentro daquela coluna numérica.



Exemplo

NUMÉRICOS

Temos por exemplo uma
tabela que armazena os
dados de um funcionário.



Exemplo

NUMÉRICOS

Esta tabela deverá ter uma
coluna para o código do
funcionário.



Exemplo

NUMÉRICOS

Esta coluna será numérica, e utilizaremos muito esta coluna, logo essas informações trafegarão pela rede.



Exemplo

NUMÉRICOS

Sendo uma coluna numérica a informação trafegará mais rapidamente pela rede.



Exemplo

NUMÉRICOS

Sendo uma coluna **numérica** a informação trafegará **mais rapidamente** pela **rede**.



Exemplo

NUMÉRICOS

Como assim

?



Exemplo

NUMÉRICOS



No cabo de rede as informações trafegam
bit a bit



Exemplo

NUMÉRICOS



8 bits formam 1 byte

Exemplo

NUMÉRICOS



1 byte equivale a 1 caractere

Exemplo

NUMÉRICOS



Um campo inteiro do tipo Tinyint* pesa
1 byte

*Veja o slide número 10



Exemplo

NUMÉRICOS



Em contrapartida, um campo `char(255)`
pesa 255 bytes



Exemplo

NUMÉRICOS



1 tinyint vs char(255)
o char neste caso é 255x mais pesado.



Exemplo

NUMÉRICOS



Qual a limitação do Tinyint ?
Se for signed, permitirá valores negativos
até -128 e positivos até 127.



Exemplo

NUMÉRICOS



Qual a limitação do Tinyint ?
Se for **unsigned**, não permitirá valores negativos
será somente de 0 a 255.



Exemplo

NUMÉRICOS



Qual a limitação do Tinyint ?
Se for **unsigned**, não permitirá valores negativos
será somente de 0 a 255.

Exemplo

NUMÉRICOS



Por isso a importância quando planejamos os datatypes de cada coluna da tabela.

Pense: Qual o maior e o menor valor que minha coluna deverá receber?

*Veja o slide número 10



Exemplo

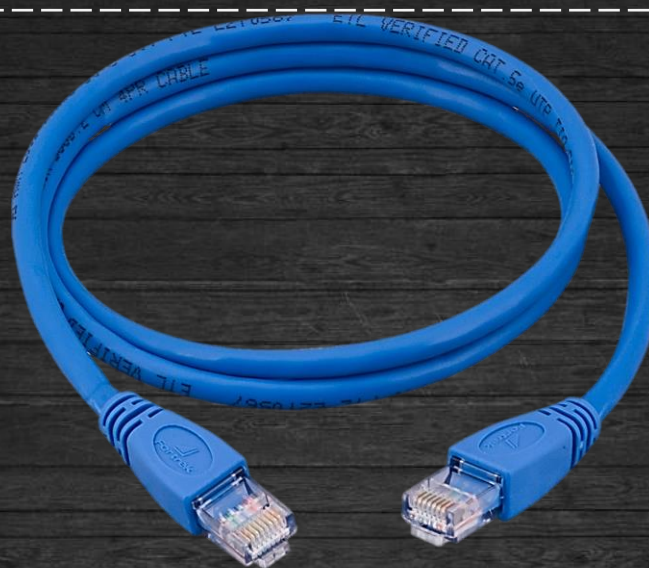
NUMÉRICOS



No exemplo em questão, a coluna código do funcionário, poderá receber um numérico inteiro que atenda ao número máximo de funcionários da empresa.

Exemplo

NUMÉRICOS



Uma empresa que tenha no máximo 500000 funcionários é um número considerável.



Exemplo

NUMÉRICOS



Neste caso um mediumint ou int resolvem o problema. Veja que Tinyint não atenderia, pois no máximo aceitaria 255 se for unsigned



Veja como seria o código de criação da tabela com esta coluna

```
create table tb_funcionario (  
    id_func          int unsigned zerofill,  
    nome_func        varchar(200),  
    email_func        varchar(250),  
    vlr_sal_func      decimal(10,2)  
);  
insert into tb_funcionario values(1, 'João' ,  
    'teste@teste.com.br' , 500.00);
```



Veja como seria o código de criação da tabela com esta coluna

```
create table tb_funcionario (  
    id_func          int signed,  
    nome_func        varchar(200),  
    email_func       varchar(250),  
    vlr_sal_func     decimal(10,2)  
);  
insert into tb_funcionario values(-10, 'João',  
    'teste@teste.com.br', 500.00);
```

Este é apenas um exemplo didático do Signed, que permite valores negativos, não seria lógico um funcionário com código negativo.



Conheça os tipos de dados

NUMÉRICOS

Numéricos de Ponto-Fixos



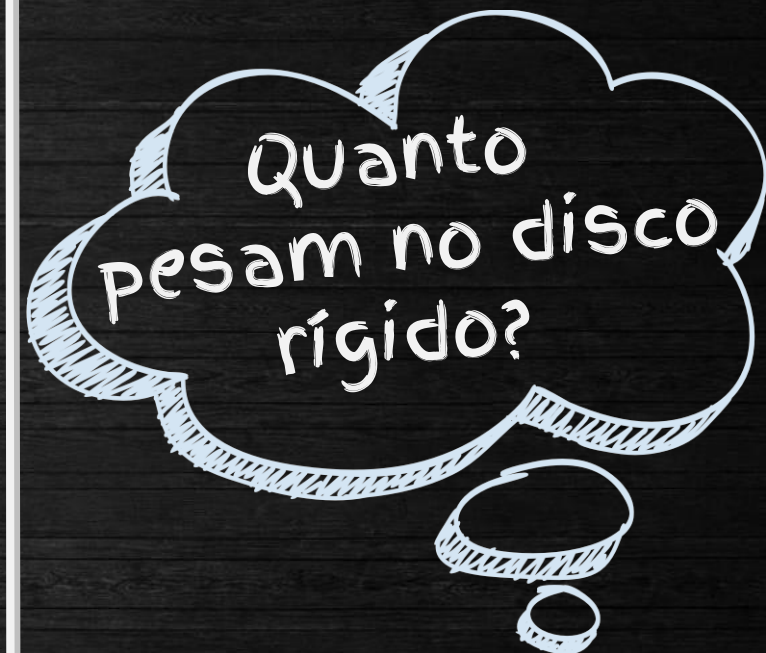
❖ Numéricos de Ponto Fixo

decimal, dec, numeric

São sinônimos, poderá utilizar qualquer um dos três.

❖ Numéricos de Ponto Fixo

decimal(10,2) 4 Bytes
decimal(18,6) 6 Bytes
decimal(12,2) 5 Bytes
decimal(30,10) 9 Bytes
decimal(65,30) 16 Bytes



Decimal

NUMÉRICOS

Como assim



Decimal

NUMÉRICOS

`decimal(10,2)`

Precisão



hcode

Decimal

NUMÉRICOS

`decimal(10,2)`

Escala



Decimal

NUMÉRICOS

`decimal(10,2)`
9999999999.99



Decimal

NUMÉRICOS

`decimal(10,2)`
9999999999.99
10 algarismos no total



Decimal

NUMÉRICOS

`decimal(10,2)`
9999999999.99
2 casas decimais



Decimal

NUMÉRICOS

`decimal(10,2)`

9999999999.99

8 números de precisão



hcode

Decimal

NUMÉRICOS

`decimal(10,2)`

9999999999.99

Até 9 números de precisão pesa 4 bytes



Decimal

NUMÉRICOS

Decimal(M,D)

M – Número **máximo** de dígitos. Limite 65

D – Número de **dígitos** do lado direito do
separador de casas decimais. Limite 30



hcode

Decimal

NUMÉRICOS

`decimal(10,2)`

9999999999.99

8 números de precisão (10-2)



hcode

Decimal

NUMÉRICOS

Dígitos de Precisão (em quantidade)	Peso no Disco Rígido (em bytes)
0	0
1-2	1
3-4	2
5-6	3
7-9	4

Decimal(10,2)

9999999999.99

8 números de precisão = 4 bytes



Decimal

NUMÉRICOS

Dígitos de Precisão (em quantidade)	Peso no Disco Rígido (em bytes)
0	0
1-2	1
3-4	2
5-6	3
7-9	4

Decimal(11,2)

9999999999.99

9 números de precisão = 4 bytes



Decimal

NUMÉRICOS

Dígitos de Precisão (em quantidade)	Peso no Disco Rígido (em bytes)
0	0
1-2	1
3-4	2
5-6	3
7-9	4

Decimal(14,5)

9999999999.99999

9 números de precisão = 4 bytes

Decimal

NUMÉRICOS

Dígitos de Precisão (em quantidade)	Peso no Disco Rígido (em bytes)
0	0
1-2	1
3-4	2
5-6	3
7-9	4

Decimal(12,2)

999999999999.99

10 números de precisão = 5 bytes

Decimal

NUMÉRICOS

Dígitos de Precisão (em quantidade)	Peso no Disco Rígido (em bytes)
0	0
1-2	1
3-4	2
5-6	3
7-9	4

Decimal(12,2)

9999999999999.99

Por que? (9 números de precisão = 4 bytes) + (1 número de precisão = 1 byte) = 5 bytes



Conheça os tipos de dados

NUMÉRICOS

Numéricos de Ponto-Flutuante



❖ Numéricos de Ponto flutuante

float, real, double

São sinônimos, poderá utilizar qualquer um dos três. Similares aos de ponto fixo, possuem algumas diferenças.

No máximo
decimal(65,30)
float(255,30)



Float Arredonda os números
decimal – Numérico exato
float – Numérico aproximado

Inserindo o valor abaixo veja o resultado

54000660040000000000922443242343.250300



Diferenças

NUMÉRICOS

Decimal(65,30)

54000660040000000000922443242343.250300000000000000000000000000

Float(255,30)

540006600400000000000000000000000000.000000000000000000000000000000



hcode

Conheça os tipos de dados

Data e Hora

Date | Time | DateTime | TimeStamp | Year



❖ Data e Hora

Date

Time

DateTime

TimeStamp

Year

❖ Data e Hora

Date	3 Bytes
Time	3 Bytes
DateTime	8 Bytes
TimeStamp	4 Bytes
Year	1 Byte



Veja como seria o código de criação da tabela com esta coluna

```
create table tb_datas (  
    id_data          int,  
    data_inicio      date,  
    data_termino     datetime,  
    ano              year  
);  
insert into tb_datas values(1, '2017-01-01',  
    '2017-01-15 12:00:00', '2017');
```

A faixa de datas suportadas para o tipo Date é: '1000-01-01' até '9999-12-31'

A faixa de valores suportado para o tipo Datetime: '1000-01-01 00:00:00' to '9999-12-31 23:59:59'



Conheça os tipos de dados

Texto

Char | VarChar | Text



❖ Texto

Char(10)	10 Bytes
Varchar(10)	1 a 10 Bytes
Text	1 a 65535 Bytes
Enum	1 ou 2 Bytes

Varchar depende do conteúdo.

Um **varchar(10)** que possui o nome **'Hcode'** contendo **5** letras irá armazenar **5 bytes**.

Veja como seria o código de criação da tabela com esta coluna

```
create table tb_funcionario (  
    id_func          int signed,  
    nome_func        varchar(200),  
    email_func        varchar(250),  
    cep_func          char(9),  
    sexo_func         enum( 'M' , 'F' )  
);  
insert into tb_funcionario values(-10, 'João' ,  
    'teste@teste.com.br' , '01311-100' , 'M');
```



MySQL 5.7

Data Type

Categorias

Numéricos | Data e Hora | Texto