

Técnico em Desenvolvimento de
sistemas
Joinville

FIESC SENAI

sc.senai.br

Banco de dados

Professor Rogério dos Santos
email:rogerio.santos@edu.sc.senai.br
Pos graduado com especialização em linguagem java

Conceitos

NUMÉRICOS DO TIPO INTEIRO

Conceitos

Datatype

- ✓ Os tipos de dados definem o formato das informações que deverão ser informadas nas colunas das tabelas.

Conceitos

Nome de colunas

- ✓ Não podem iniciar com números ou caracteres especiais. O único caracter permitido é _ (**Underline**).
- ✓ Para os nomes de colunas, é possível conter números no meio ou no fim.

Conceitos

Exemplo de nomes de colunas

id_func

nome_func

email_func

vlr_sal_func



Os nomes das colunas não podem ter espaços e nem iniciar com números, caracteres especiais

Conceitos

Tipos de dados numéricos inteiros

Tipo	Peso no disco rígido
tinyint	1 byte
smallint	2 byte
mediumint	3 byte
int	4 byte
bigint	8 byte

Tipo	Menores e maiores valores aceitos
tinyint	-128 a 127
smallint	-32768 a 32767
mediumint	-8388608 a 8388607
int	-2147483648 a 2147483647
bigint	-9223372036854775808 a - 9223372036854775807

Conceitos

Os tipos numéricos são divididos entre:

signed - É o padrão. Permite números negativos.

unsigned - Não permite números negativos.

Conceitos

O que significa menores e maiores valores aceitos ?

Conceitos

Exemplo uma tabela que armazena os dados do **funcionário**.

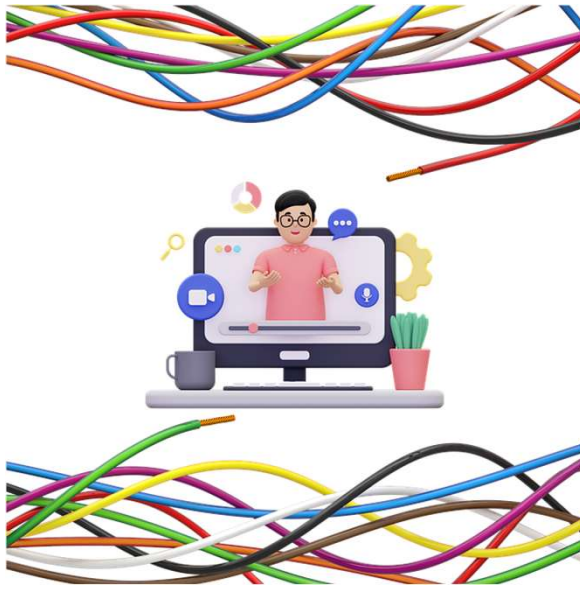
Esta tabela deverá ter uma coluna para armazenar o código do **funcionário**.

Conceitos

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

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

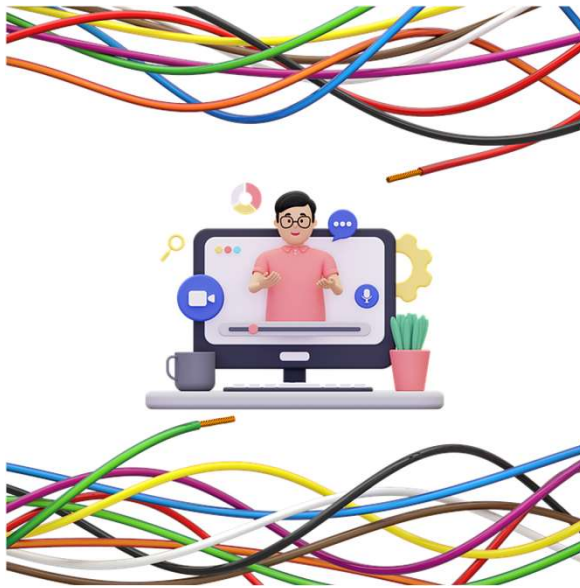
Conceitos



No cabo de rede as informações trafegam **bit a bit**

8 bits formam um byte

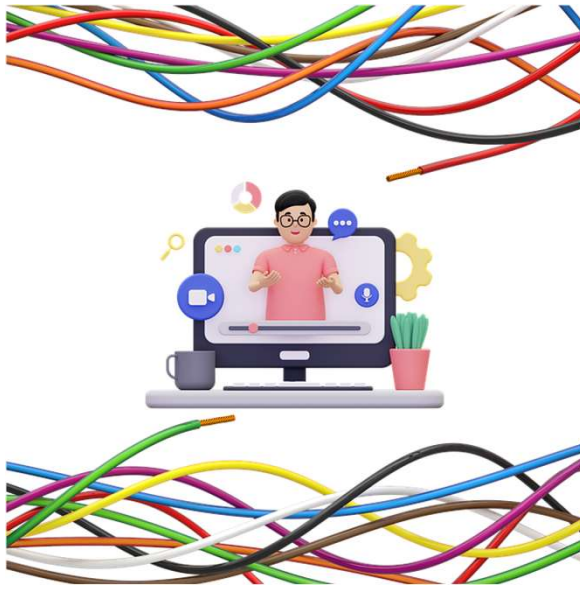
Conceitos



1 byte equivale a um caracter.

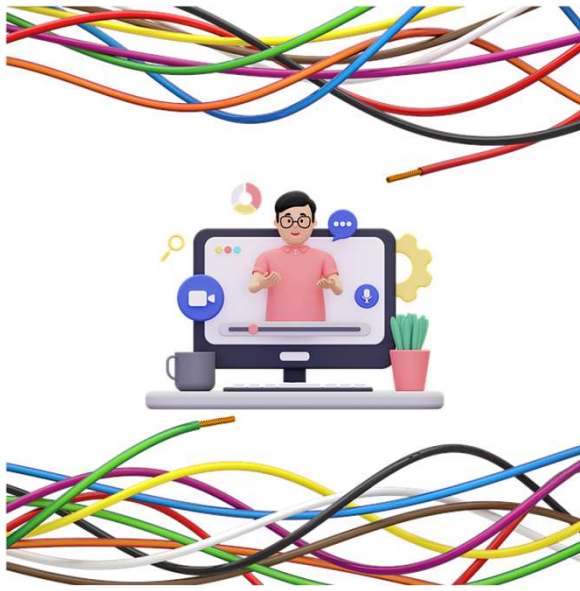
1 campo inteiro do tipo Tinyint pesa um byte de memória

Conceitos



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

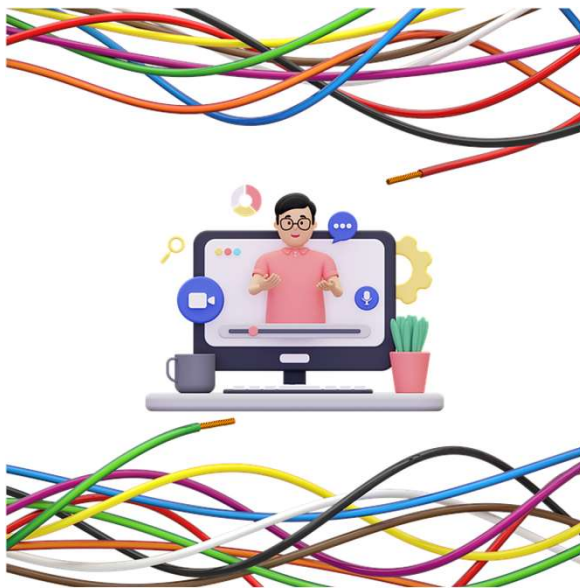
Conceitos



Tynint x char(255)

**Neste caso o char é 255
vezes mais pesado.**

Conceitos

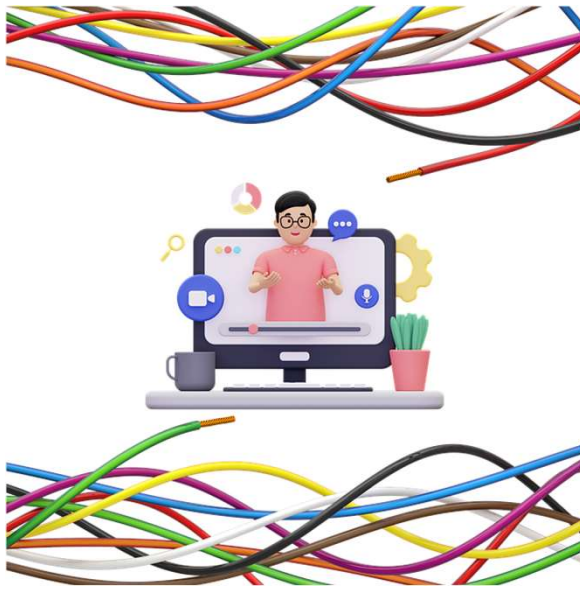


Qual a limitação do Tynint ?

Se for **signed** permitirá valores negativos até **-128** e positivos até **127**?

Se for **unsigned** não permitirá valores negativos, será somente de **0 a 255**.

Conceitos

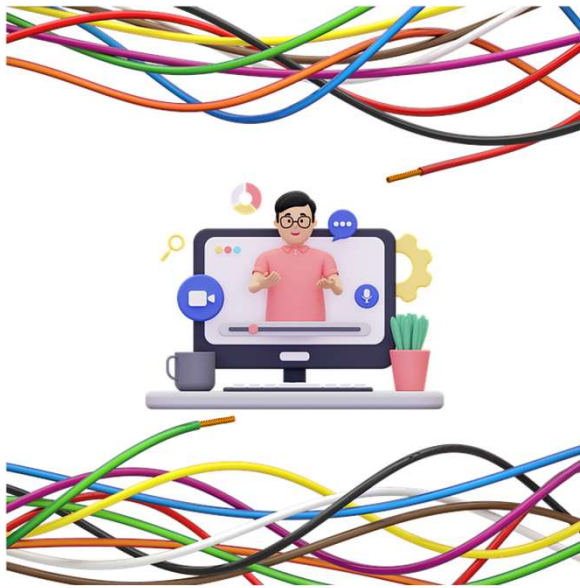


Por isso a importância de planejar de qual **datatype** cada coluna da tabela deverá receber

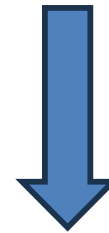


Pense: Qual o maior e menor valor que a coluna da tabela deverá receber ?

Conceitos



Uma empresa que tenha no máximo 500000 funcionários é um tamanho considerável



Um campo do tipo **mediumint** ou **int** resolve o problema

Conceitos

Exemplo da criação da tabela de funcionários

```
CREATE TABLE tb_funcionario(  
    id_func int unsigned zerofill,  
    nome_func varchar(40),  
    email varchar(100),  
    vlr_salario decimal(10,2),  
  
    constraint pk_funcionario primary key(id_func)  
  
)
```

Conceitos

Exemplo de inserção de registros da tabela de funcionários

```
INSERT INTO tb_funcionario values(1,"João","joao@gmail.com",2000.00)
```

Resultado após a inserção

id_func	nome_func	email	vlr_salario
0000000001	João	joao@gmail.com	2000.00

O uso de zerofill ao lado da coluna id_func permite colocar zeros à esquerda

Conceitos

Outro exemplo com uso de **zerofill com campo do tipo int sendo de 4 dígitos**

```
CREATE TABLE tb_empresa(  
  id int(4) unsigned zerofill,  
  nm_razaosocial varchar(40),  
  email varchar(100),  
  constraint pk_empresa primary key(id)  
  
);
```

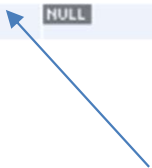
Conceitos

Inserção na tabela tb_empresa

```
INSERT INTO tb_empresa values(1,"Software LTDA","software@gmail.com");
```

Resultado da inserção na tabela tb_empresa

	id	nm_razaosocial	email
▶	0001	Software LTDA	software@gmail.com
◀	NULL	NULL	NULL



O uso de zerofill no campo do tipo int(4)
com 4 dígitos

Conceitos

Exemplo da criação da tabela tb_cliente com a coluna id, sendo do tipo **int** e **signed**

```
CREATE TABLE tb_cliente(  
    id int(4) signed,  
    nm_cliente varchar(40),  
    email varchar(100),  
    constraint pk_cliente primary key(id)  
);
```


Conceitos

Exemplo da criação da tabela tb_cliente com a coluna id, sendo do tipo **int** e **signed**

```
CREATE TABLE tb_cliente(  
    id int(4) signed,  
    nm_cliente varchar(40),  
    email varchar(100),  
    constraint pk_cliente primary key(id)  
);
```

Conceitos

Exemplo de inserção de registros na tabela
tb_cliente

```
INSERT INTO tb_cliente values(-10,"Pedro","pedro@gmail.com");
```

Não seria lógico colocar um valor de um campo para código como negativo. Este exemplo é simplesmente didático

Conceitos

NUMÉRICOS DO PONTO FIXO

Conceitos

Precisão

Escala

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

Conceitos

decimal(10,2)



999999999.99

Conceitos

NUMÉRICOS DE PONTO FLUTUANTE

Conceitos

Float, real e double

No máximo

Diferenças {
Decimal(65,30)
Float(255,30)

Conceitos

TIPOS DATA E HORA

Date, Time, DateTime, TimeStamp e Year

Conceitos

Date – 3 bytes

Time - 3 bytes

DateTime– 8 bytes

TimeStamp 4 bytes

Year 1 byte

Conceitos

Código para criação da tabela com datas

```
CREATE TABLE datas(  
    id_data int,  
    data_inicio date,  
    data_terminio datetime,  
    ano year,  
    CONSTRAINT fk_datas PRIMARY KEY(id_data)  
)
```

Conceitos

Texto


Char, Varchar, Text, Enum

Conceitos

Código para criação da inserção de registros

```
INSERT INTO datas values(1,"2024-09-05","2020-09-6 19:30","2024");
```

Resultado da inserção de registros



	id_data	data_inicio	data_terminio	ano
▶	1	2024-09-05	2020-09-06 19:30:00	2024
★	NULL	NULL	NULL	NULL

Conceitos

Char(10) – 10 bytes

Varchar(10) - 10 bytes

Text– 1 a 65535 bytes

Enum 1 a 2 bytes

Conceitos



O tipo de dado **varchar**, depende do conteúdo. Um **varchar(10)** que possui o nome **Banco**, irá armazenar **5 bytes**.

Conceitos

Código para criação da tabela com tipo de dado texto

```
CREATE TABLE tb_funcionario(  
    id_func int unsigned zerofill,  
    nome_func varchar(40),  
    email varchar(100),  
    cep char(9),  
    observacao text(50),  
    genero enum('M','F'),  
  
    constraint pk_funcionario primary key(id_func)  
  
);
```

Conceitos

Código para inserção de registro

```
INSERT INTO tb_funcionario  
values(1,"João","joao@gmail.com","89221-008","Funcionario com boa experiência na funcao","M");
```

Resultado da inserção de registro

	id_func	nome_func	email	cep	observacao	genero
▶	0000000001	João	joao@gmail.com	89221-008	Funcionario com boa experiência na funcao	M
•	NULL	NULL	NULL	NULL	NULL	NULL

Referência bibliográfica

Milani, André. Mysql- Guia do programador. São Paulo.Novatec. 2010.389 p.

Guimarães , Célio Cardoso. Fundamentos de banco de dados.Unicamp.2010.262 p.