



Fundamentos do banco de dados SQL SERVER

BANCO



Por que um banco?

Por que um banco?



Local Adequado



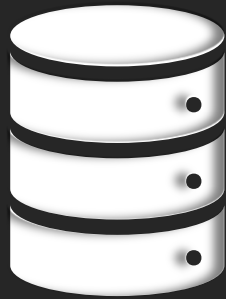
Segurança de Acesso
aos Dados



Confiabilidade



Disponibilidade



Organizando Informações



Tirar as Roupas
do Varal

ALTA



Tarefa e
Prioridade

Estudar SQL
Server ✓

~~Instalação e~~
~~logins~~



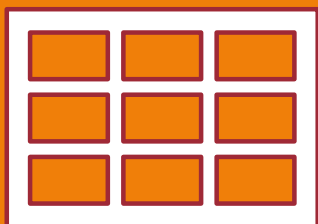
Tarefa +
Descrição +
Conclusão

Reunião com
Cliente

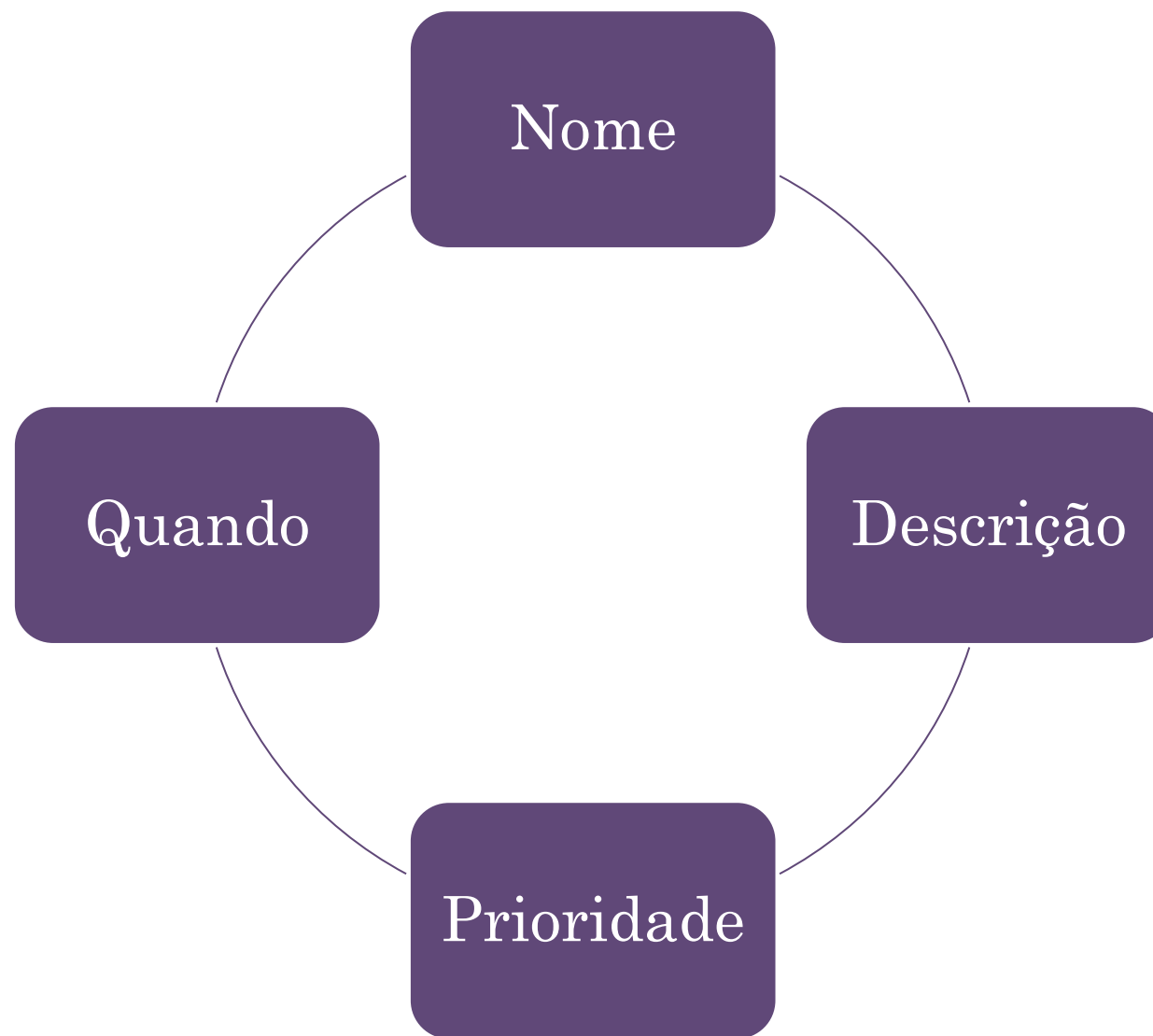
amanhã



Tarefa + Quando



Tarefa



Você deseja **informatizar** suas atividades. Você cria então uma planilha (tabela) chamada **tarefa.xlsx**

Nome	Descrição	Prioridade	Quando Concluído
Tirar as Roupas do Varal		ALTA	
Estudar SQL Server	Instalação e logins		Sim
Reunião com Cliente			Amanhã

Como você deseja
estudar **SQL** e quem
sabe montar um sistema
para grandes equipes...
você decide criar um
banco chamado
Bluemine que irá conter
todos os dados de sua
planilha **tarefas.xlsx**

Banco **Bluemine**





Criar Banco Bluemine



Fundamentos do banco de dados

SQL SERVER

TABELA

Banco BlueMine

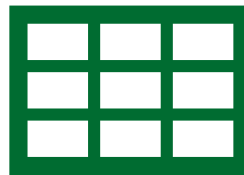


Tabela Tarefa

Tabela Tarefa

primeiros passos

Coluna	Tipo
Nome	Texto
Descricao	Texto
Prioridade	Texto ou Numero?

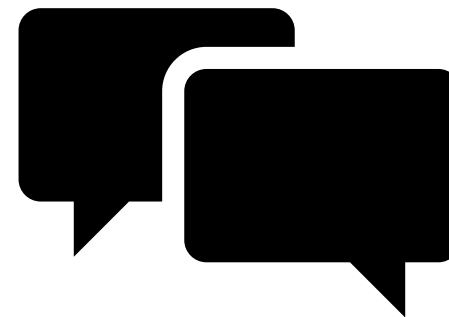


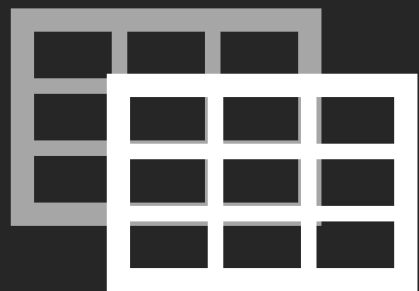
Tabela Tarefa

primeiros passos

Coluna	Tipo
Nome	Varchar(50)
Descricao	Varchar(200)
Prioridade	int

Prioridade





Criar Tabela Tarefa

Fundamentos do banco de dados SQL SERVER

Colunas



COLUNAS



Visualizar Dados

SELECT * From tarefas



Alimentando Tabelas

INSERT – ADICIONAR VALORES

INSERT INTO tarefas

(nome, descricao, prioridade)

VALUES

('Estudar', 'Estudar SQL', 1)

INSERT – ADICIONAR VALORES

INSERT INTO tarefas

(nome, descricao, prioridade)

VALUES

('Estudar', 'Estudar SQL', 1)

INSERT – ADICIONAR VALORES

INSERT INTO tarefas

(nome, descricao, prioridade)

VALUES

('Estudar', 'Estudar SQL', 1)

INSERT – ADICIONAR VALORES

INSERT INTO tarefas

(nome, descricao, **prioridade**)

VALUES

('Estudar', 'Estudar SQL', **1**)

INSERIR MÚLTIPLOS VALORES

```
INSERT INTO tarefas  
(nome, descricao, prioridade)
```

```
VALUES
```

```
( 'SQL', 'INSERT, SELECT, UPDATE, DELETE', 1 ),
```

```
( 'Reunião com Cliente', '', 2 ),
```

```
( 'Inciar academia', '', 1 ),
```

```
( 'Mini projeto', 'Projeto Bluemine', 4 );
```

INSERIR MÚLTIPLOS VALORES

```
INSERT INTO tarefas  
(nome, descricao, prioridade)  
VALUES
```

```
( 'SQL', 'INSERT, SELECT, UPDATE, DELETE', 1 ),  
( 'Reunião com Cliente', '', 2 ),  
( 'Inciar academia', '', 1 ),  
( 'Mini projeto', 'Projeto Bluemine', 4 );
```


Ao adicionar um registro

- Os valores das colunas não especificadas serão salvas como **NULL**

```
insert Tarefa(nome)  
values ('Nova Tarefa Sem Desc ou Prioridade')
```

nome	descricao	prioridade
Nova Tarefa Sem Desc ou Prioridade	NULL	NULL

Ao adicionar um
novo **registro**
preciso sempre
digitar todas as
colunas?

VALORES DEFAULT

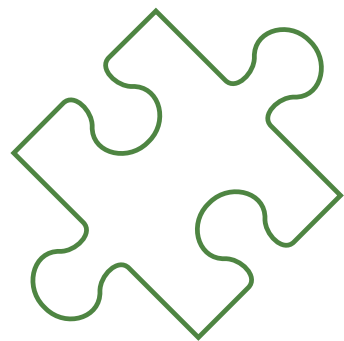


Valor Default

```
create table Tarefa (  
    nome varchar(50),  
    descricao varchar(200),  
    prioridade int default(3)  
)
```

*BOA PRÁTICA:
especifique
todos os campos
que serão
inseridos*



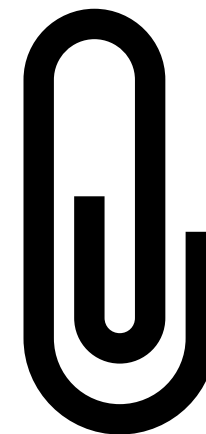


Adicionando
novos campos



Tabela Tarefa

Coluna	Tipo
Nome	Varchar(50)
Descricao	Varchar(200)
Prioridade	Varchar(5)
Quando	date
Concluído	Bit



Date

Tipos de Dados

ano-mês-dia

2018-05-30

2018-01-01

2019-02-15



Bit

TIPOS DE DADOS



wiseGEEK



Posso ter mais de um **tipo**
de dado por **coluna**?

RESPOSTA: Uma coluna suporta apenas um tipo de dado

Inserir uma tarefa

```
insert tarefa  
  (nome, descricao, prioridade, quando, concluido)  
values  
  ('Estudar SQL', '', 1, '20180501' , 0)
```

Inserir uma tarefa sem data

```
insert tarefa
(nome, descricao, prioridade, quando, concluido)
values
('Estudar SQL', '', 1, NULL, 0)
```

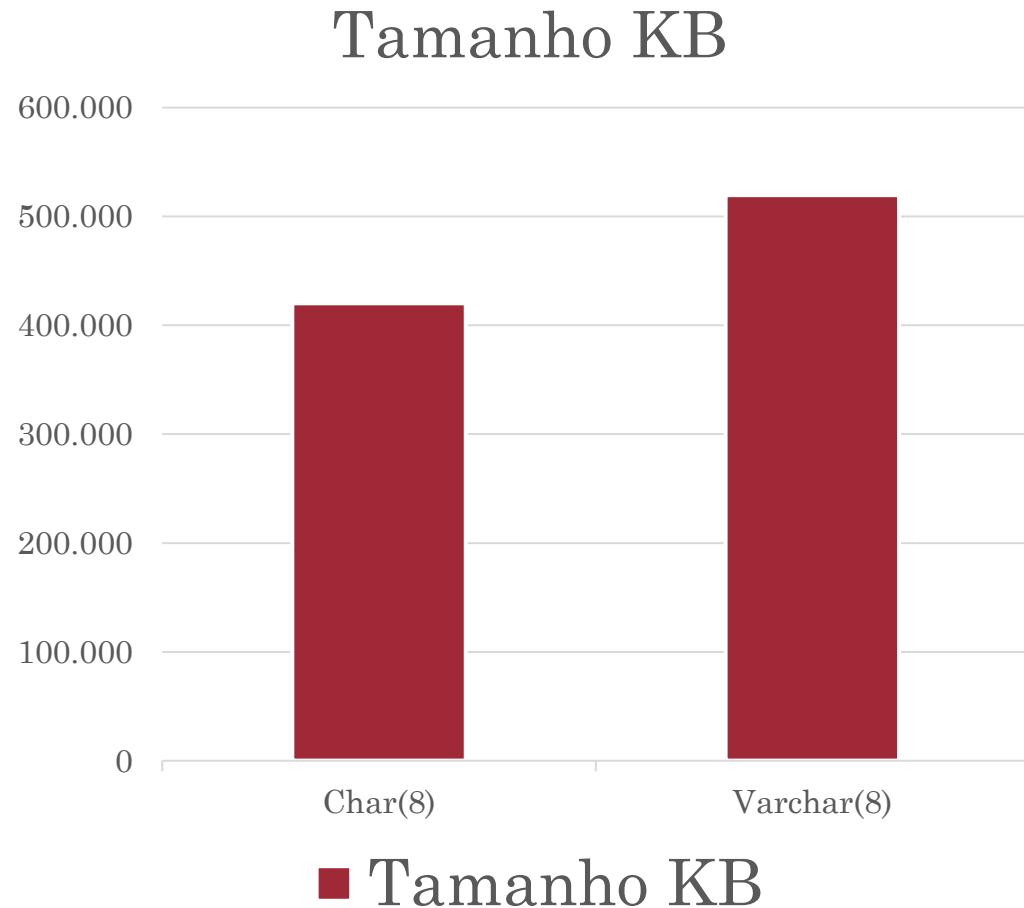

Vantagens NOT NULL

- Consistência dos dados
 - Na tabela cliente, o nome é obrigatório. Não faz sentido permitir informações incompletas.
- Facilita manipulação de dados
 - *Name* != 'Ana' and *Name* is not null
- Ocupam menos espaço

char(8) vs varchar(8)

25.000.000 linhas de 8 caracteres

varchar(8) possui
99.312KB a mais.



char(8) vs varchar(8)

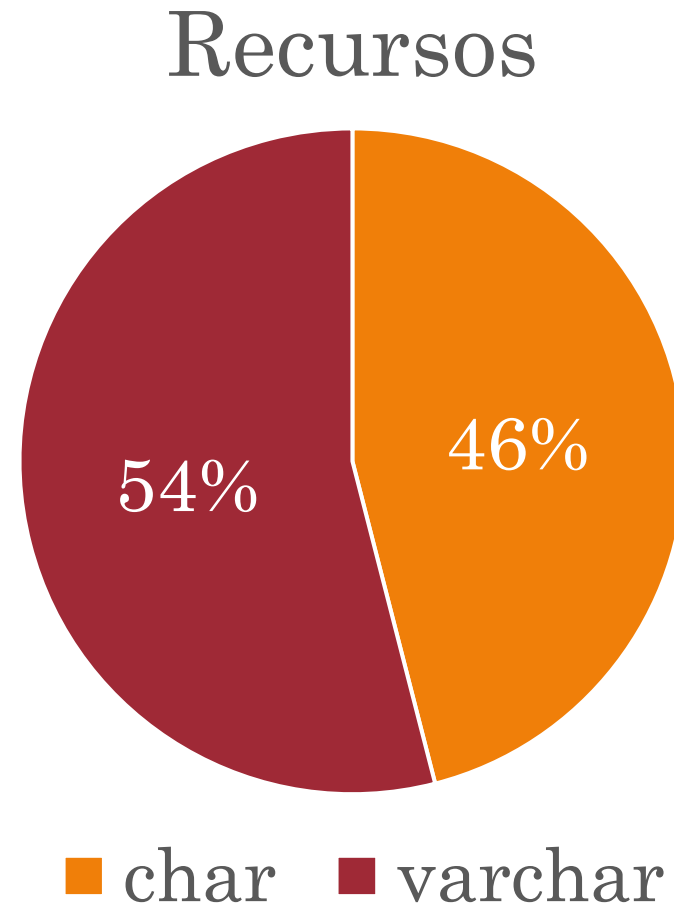
25.000.000 linhas de 8 caracteres (sem índice)

Varchar consome mais I/O devido ao seu tamanho. Portanto, tende a ser mais lento nas operações.

Select *

From Tabela

Where Coluna = 'Valor'





Sparse Columns

Otimiza o espaço consumido em colunas quando a maioria dos registros são nulos.

Performance e Tipos de dados

- <https://sqlperformance.com/2012/09/t-sql-queries/data-type-choice>
- <https://www.sqlshack.com/sql-server-data-type-conversion-methods-performance-comparison/>
- <https://www.sentryone.com/white-papers/data-type-choice-affects-database-performance>
- <http://www.sparkhound.com/blog/five-scariest-sql-server-data-types>
- <https://www.sentryone.com/white-papers/data-type-choice-affects-database-performance>



Como identificar minha tarefa?

CHAVE PRIMÁRIA

Identificador

Coluna	Tipo
TarefaId	int
Nome	Varchar(50)
Descricao	Varchar(200)
Prioridade	int
Quando	date
Concluído	Bit



Sequencial
1, 2, 3, 4

Identificador

```
create table Tarefa (  
  TarefaId int identity(1,1),  
  Nome varchar(50) NOT NULL,  
  Descricao varchar(200) NOT NULL,  
  Prioridade int NOT NULL default(3),  
  Concluido bit NOT NULL  
)
```

Sequencial
1, 2, 3, 4

Chaves primárias *Resumo*



Identificador



Coluna	Tipo
TarefaId	int
Nome	Varchar(50)
Descricao	Varchar(200)
Prioridade	int
Quando	date
Concluído	Bit



Sequencial
1, 2, 3, 4

Vantagens da **Chave primária**

- Identifica a tarefa
- **Garante que a tarefa é única**
 - *Não permite duplicados*
- Performance nas consultas

Identificador

`create table Tarefa (`
 TarefaId `int primary key identity(1,1),`
 Nome `varchar(50) NOT NULL,`
 Descricao `varchar(200) NOT NULL,`
 Prioridade `int NOT NULL default(3),`
 Concluido `bit NOT NULL`
`)`

Chave primária



GitHub

<https://github.com/emanuelfqueiroz>



Linkedin

<https://www.linkedin.com/in/emanuelfeliciano/>

