

Marcelo Abib Cardoso

SQLServer – Constraints

Olá Pessoal,

Neste artigo estaremos falando um pouco mais de constraints e como colocar valores padrões em colunas.

Segue:

- O que são constraints
- Onde elas são utilizadas
- Tipos de constraints

O que são constraints?

Constraints são restrições feitas para as colunas nas tabelas contendo diversos tipos, na qual falaremos adiante.

Onde elas são utilizadas?

Elas são utilizadas na criação de uma tabela ou mesmo junto com a keyword ALTER TABLE, onde podemos adicionar ou remover constraints.

Tipo de constraints:

– Primary key(PK)

Já vimos anteriormente no artigo criação de tabelas(Simples).

Ex:

```
CREATE TABLE Exemplo
```

```
(
```

```
ExemploId int not null Identity(1,1),
```

```
Campo2 int not null,
```

```
CONSTRAINT PK_Exemplo_ExemploId PRIMARY KEY(ExemploId)
```

```
)
```

– Unique(UQ)

Cria uma restrição para a coluna, onde não pode ter dois valores iguais.

Ex:

```
CREATE TABLE Newsletter  
  
(  
  
NewsletterId int not null IDENTITY(1,1) CONSTRAINT PK_Newsletter_NewsletterId PRIMARY  
KEY,  
  
Email nvarchar(64) not null CONSTRAINT UQ_Newsletter_Email UNIQUE  
  
)
```

Ao inserir dois e-mails iguais ele vai gerar um erro de UNIQUE KEY, dizendo que você está duplicando o registro.

– Foreign key(FK)

Ainda não vimos o que é um foreign key(Chave Estrangeira), será um próximo artigo, mas resumindo cria-se uma referência com o índice(PK) de uma outra tabela.

Ex:

```
CREATE TABLE Funcao  
  
(  
  
FuncaoId int not null IDENTITY(1,1) CONSTRAINT PK_Funcao_FuncaoId PRIMARY KEY,  
  
Nome nvarchar(32) not null  
  
)  
  
CREATE TABLE Funcionario  
  
(  
  
FuncionarioId int not null IDENTITY(1,1) CONSTRAINT PK_Funcionario_FuncionarioId PRIMARY  
KEY,  
  
Nome nvarchar(64) not null,  
  
FuncaoId int not null,  
  
CONSTRAINT FK_Funcionario_FuncaoId FOREIGN KEY(FuncionarioId)  
REFERENCES Funcao(FuncaoId)  
  
)
```

Neste exemplo temos a tabela Funcao que pode ser Gerente, Desenvolvedor e etc.

e na tabela Funcionario colocamos o Nome, ex Marcelo e a funcao, porem na funcao nao vamos colocar Desenvolvedor, e sim colocamos a referencia que ela é na tabela de funcao, Ex: 2

Visualmente ficaria assim

Tabela: Funcao

FuncaoId Nome

1 Gerente

2 Desenvolvedor

Tabela: Funcionario

FuncionarioId Nome FuncaoId

1 Marcelo 2

Se colocarmos um FuncaoId que não existe na tabela de Funcao dará erro de Foreign Key.

– Check(CK)

Check, é uma constraint que especifica que condição a coluna precisa para salvar um registro.

Ex:

Na tabela Cliente, temos o campo SEXO e nele só pode ser inserido F ou M, então criamos uma constraint de check, só permitindo essas duas letras e evitando que seja inserido A, B, C.

```
CREATE TABLE Cliente
```

```
(
```

```
  ClienteId int not null IDENTITY(1,1) CONSTRAINT PK_Cliente_ClienteId PRIMARY KEY,
```

```
  Nome nvarchar(64) not null,
```

```
  Sexo char(1) not null CONSTRAINT CK_Cliente_Sexo CHECK(Sexo = 'M' or Sexo = 'F')
```

```
)
```

Ao ser inserido A, B, C ele não permitirá e dará erro na constraint CK_Cliente_Sexo, pois ela só aceita M ou F.

E para finalizar os tipos de constraints e o artigo falaremos da Constraint **DEFAULT(DF)**

Essa constraint é responsável por colocar valores padrões nas colunas.

Ex: DataCadastro eu posso colocar um valor padrão que seria a data de hoje.

```
CREATE TABLE Teste
```

```
(
```

```
  Nome nvarchar(64) not null,
```

```
  DataCadastro datetime not null CONSTRAINT DF_Testes_DataCadastro DEFAULT getDate()
```

```
)
```