



*BANCO DE DADOS*

**TRIGGERS**

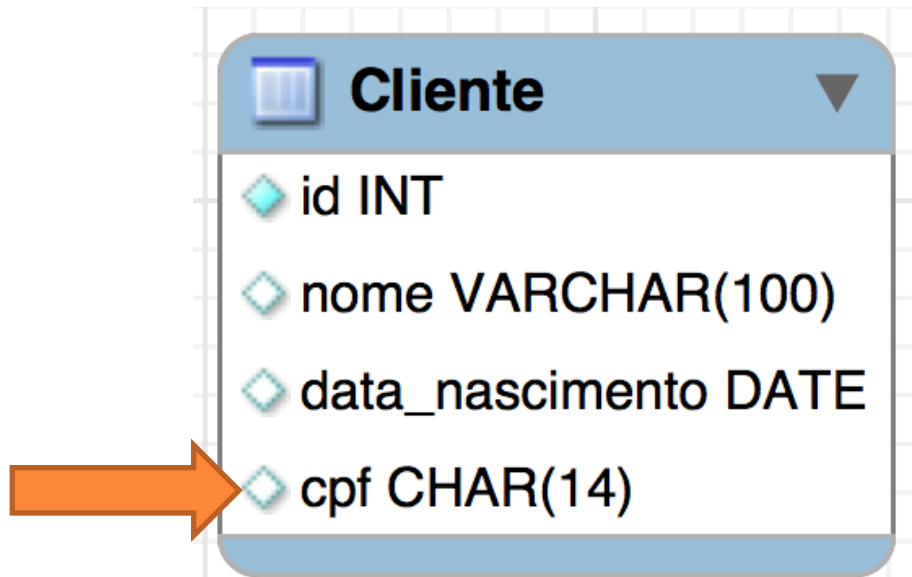
**VALIDAÇÃO DE DADOS**

**CORREÇÃO DOS EXERCÍCIOS**

**Prof. Fabiano Papaiz**  
**IFRN**

# VALIDAÇÃO – CPF DO CLIENTE

- Campo **CPF** do Cliente



# VALIDAÇÃO – CPF DO CLIENTE

## ○ Código SQL:

```
declare @cpf varchar(14)
declare @numeros_validos char(10) = '1234567890'

--guarda o CPF informado
select  @cpf = cpf from inserted

--remove todos os espaços em branco do @cpf
set @cpf = REPLACE(@cpf, ' ', '')
```

## ○ Passos:

- Declara variáveis para guardar o CPF informado e para definir os números válidos (essa última para evitar o problema da função *isnumeric* quando passamos +, e etc)
- Obtém o CPF informado
- Remove todos os espaços em branco do CPF

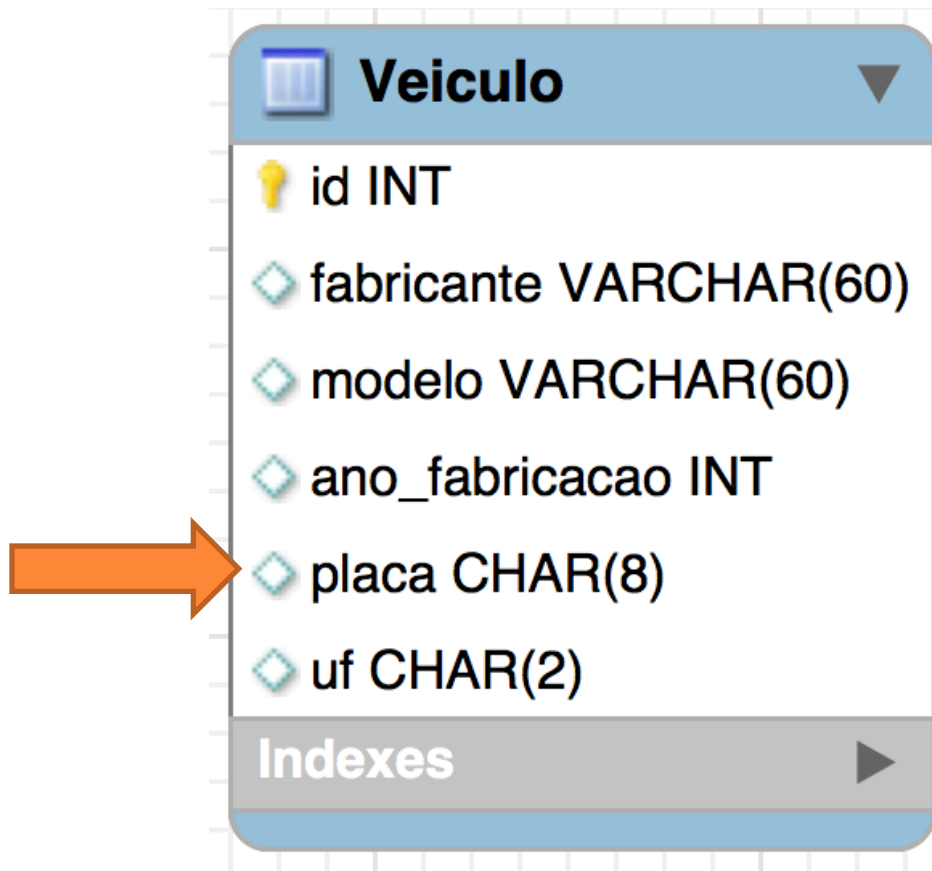
# VALIDAÇÃO – CPF DO CLIENTE

- Verifica se o CPF está no formato 999.999.999-99:

```
--verifica se o CPF esta no formato 999.999.999-99
if (len(@cpf) <> 14)
  or( charindex(substring(@cpf, 1, 1), @numeros_validos) = 0 )
  or( charindex(substring(@cpf, 2, 1), @numeros_validos) = 0 )
  or( charindex(substring(@cpf, 3, 1), @numeros_validos) = 0 )
  or( substring(@cpf, 4, 1) <> '.' )
  or( charindex(substring(@cpf, 5, 1), @numeros_validos) = 0 )
  or( charindex(substring(@cpf, 6, 1), @numeros_validos) = 0 )
  or( charindex(substring(@cpf, 7, 1), @numeros_validos) = 0 )
  or( substring(@cpf, 8, 1) <> '.' )
  or( charindex(substring(@cpf, 9, 1), @numeros_validos) = 0 )
  or( charindex(substring(@cpf, 10, 1), @numeros_validos) = 0 )
  or( charindex(substring(@cpf, 11, 1), @numeros_validos) = 0 )
  or( substring(@cpf, 12, 1) <> '-' )
  or( charindex(substring(@cpf, 13, 1), @numeros_validos) = 0 )
  or( charindex(substring(@cpf, 14, 1), @numeros_validos) = 0 )
begin
  rollback transaction
  raiserror('CPF incorreto', 0, 0)
end
```

# VALIDAÇÃO – PLACA DO VEÍCULO

- Campo **PLACA** do Veículo



# VALIDAÇÃO – PLACA DO VEÍCULO

- Código SQL:

```
declare @placa varchar(8)
declare @numeros_validos char(10) = '1234567890'
declare @letras_validas char(26) = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'

--guarda a placa informada
select @placa = placa from inserted

--remove todos os espaços em branco da placa
set @placa = REPLACE(@placa, ' ', '')
```

- Passos:

- Declara variáveis para guardar a placa informada e para definir as letras e números válidos
- Obtém placa informada
- Remove todos os espaços em branco da placa

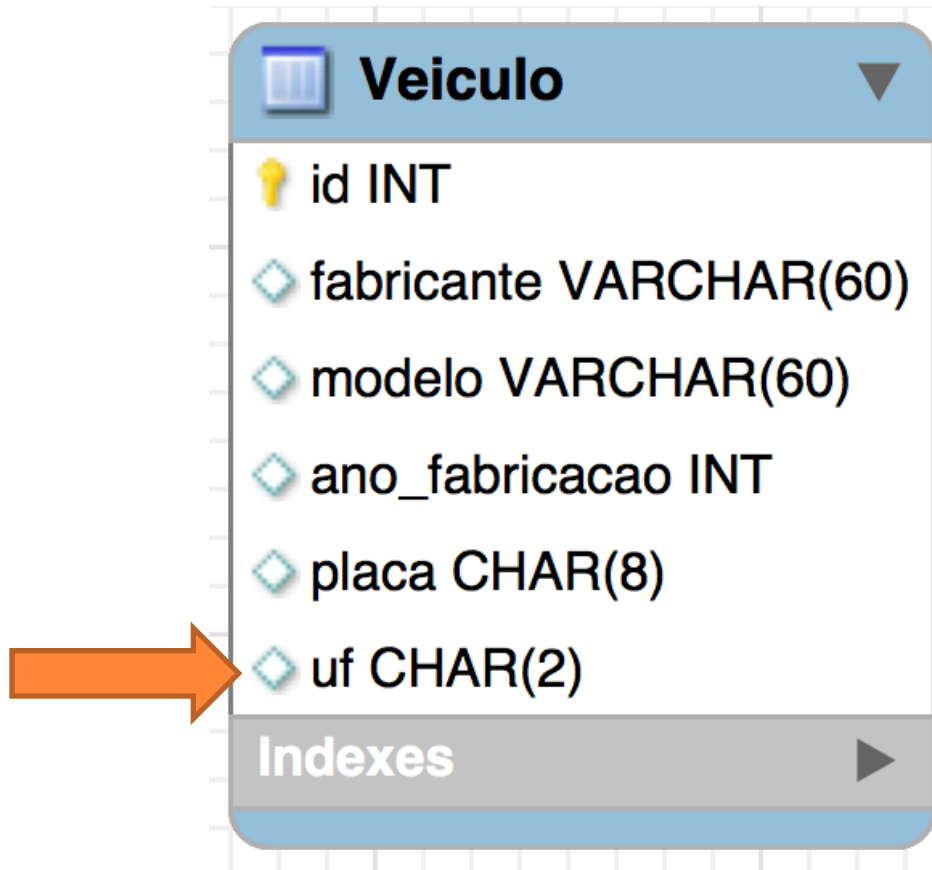
# VALIDAÇÃO – PLACA DO VEÍCULO

- Verifica se a placa está no formato **AAA-9999**:

```
--verifica se a placa esta no formato AAA-9999
if (len(@placa) <> 8)
  or( charindex(substring(@placa, 1, 1), @letras_validas) = 0 )
  or( charindex(substring(@placa, 2, 1), @letras_validas) = 0 )
  or( charindex(substring(@placa, 3, 1), @letras_validas) = 0 )
  or( substring(@placa, 4, 1) <> '-' )
  or( charindex(substring(@placa, 5, 1), @numeros_validos) = 0 )
  or( charindex(substring(@placa, 6, 1), @numeros_validos) = 0 )
  or( charindex(substring(@placa, 7, 1), @numeros_validos) = 0 )
  or( charindex(substring(@placa, 8, 1), @numeros_validos) = 0 )
begin
  rollback transaction
  raiserror('Placa incorreta', 0, 0)
end
```

# VALIDAÇÃO – UF DO VEÍCULO

- Campo **UF** do Veículo





# VALIDAÇÃO – UF DO VEÍCULO

- Verifica se a UF informada é umas das 27 existentes:

```
--guarda a uf informada|
declare @uf varchar(2)
select  @uf = uf from inserted

--remove todos os espaços em branco da UF
set @uf = REPLACE(@uf, ' ', '')

--verifica a uf
if (len(@uf) <> 2)
    or(@uf not in ('AC', 'AL', 'AP', 'AM', 'BA', 'CE', 'DF', 'ES'
                  , 'GO', 'MA', 'MT', 'MS', 'MG', 'PA', 'PB', 'PR'
                  , 'PE', 'PI', 'RJ', 'RN', 'RS', 'RO', 'RR', 'SC'
                  , 'SP', 'SE', 'TO')) )
begin
    rollback transaction
    raiserror ('UF invalida', 0, 0)
end
```



RESOLUÇÃO UTILIZANDO  
*EXPRESSÕES REGULARES*  
ATRAVÉS DO  
OPERADOR *LIKE*

# VALIDAÇÃO – EXPRESSÕES REGULARES

- Utilizando *Expressões Regulares* através do operador de comparação *LIKE* do *SQL Server*, podemos verificar mais facilmente (usando menos código) se uma *string* está de acordo com um determinado formato ou padrão
- Geralmente são utilizadas para validar entradas de dados ou fazer busca e extração de informações em textos
- Hoje em dia, as expressões regulares estão disponíveis na maioria das linguagens de programação existentes

# VALIDAÇÃO – **PLACA** DO VEÍCULO

```
--guarda a placa informada
declare @placa varchar(8)
select  @placa = placa from inserted

--verifica se a placa estah no formato AAA-9999
if @placa not like '[A-Z][A-Z][A-Z]-[0-9][0-9][0-9][0-9]'
begin
    rollback transaction
    raiserror('Placa incorreta', 0, 0)
end
```

- *OBS: não precisamos nem verificar se o tamanho da placa informada é diferente de 8 caracteres*

# VALIDAÇÃO – CPF DO CLIENTE

```
declare @cpf varchar(14)|  
  
--guarda o CPF informado  
select @cpf = cpf from inserted  
  
--verifica se o CPF esta no formato 999.999.999-99  
if @cpf not like '[0-9][0-9][0-9].[0-9][0-9][0-9].[0-9][0-9][0-9]-[0-9][0-9]'  
begin  
    rollback transaction  
    raiserror('CPF invalido*', 0, 0)  
end
```

- *OBS: não precisamos nem verificar se o tamanho do CPF informado é diferente de 14 caracteres*



**FIM**