

Microsoft SQL Server 2016 (T-SQL)

Aula 02

Professor Anderson Henrique

Assuntos tratados nessa aula:

- 01 – Criação de um banco de dados
- 02 – SQL Constraints (Restrições)
- 03 – Normalização e Desnormalização

Professor Anderson Henrique

01 – Criando um banco de Dados

Para criar um banco de dados, utilizamos o comando DDL com a seguinte sintaxe:

```
CREATE DATABASE nome_do_banco  
ON PRIMARY(  
    NAME = nome_do_banco,  
    FILENAME = 'o caminho onde o arquivo deve ser salvo',  
    SIZE = tamanho MB,  
    MAXSIZE = tamanhomáximo MB,  
    FILEGROWTH= taxacrescimento %  
)
```

Vamos criar o seguinte banco de dados:

```
CREATE DATABASE db_Biblioteca  
ON PRIMARY(  
    NAME = 'db_Biblioteca',  
    FILENAME = 'C:\sql\db_Biblioteca.MDF',  
    SIZE=6MB,  
    MAXSIZE=15MB,  
    FILEGROWTH=10%  
);
```

Professor Anderson Henrique

Comando DDL utilizando a Sintaxe breve:

CREATE DATABASE nome_do_banco;

CREATE DATABASE db_Biblioteca;

USE

Comando instrui o SGDBR a utilizar o banco de dados especificado para executar os comandos

Sintaxe básica:

USE banco_de_dados;

sp_helpdb

Informa o tamanho, taxa de crescimento, e o local do banco de dados

DROP DATABASE

O comando elimina o banco de dados especificado

Sintaxe básica:

DROP DATABASE banco_de_dados;

02 – SQL Constraints (Restrições)

Restrições são regras aplicadas nas colunas de uma tabela

São usadas para limitar os tipos de dados que são inseridos

Podem ser especificadas no momento de criação da tabela (CREATE) ou após a tabela ter sido criada (ALTER)

Principais Constraints:

NOT NULL

UNIQUE

PRIMARY KEY

FOREIGN KEY

CHECK

DEFAULT

NOT NULL

A constraint que impõe a uma coluna a NÃO aceitar valores NULL

A constraint obriga um campo a sempre possuir um valor

Deste modo, não é possível inserir um registro (ou atualizar) sem entrar com um valor neste campo

UNIQUE

A restrição identifica de forma única cada registro em uma tabela de banco de dados

As constraints UNIQUE e PRIMARY KEY garantem a unicidade em uma coluna ou conjunto de colunas

Uma constraint PRIMARY KEY automaticamente possui uma restrição UNIQUE definida

Você pode ter várias constraints UNIQUE em uma tabela, mas apenas uma PRIMARY KEY por tabela

PRIMARY KEY

A PRIMARY KEY identifica de forma única cada registro em uma tabela de banco de dados

Chaves Primárias devem conter valores únicos

Uma coluna de chave primária não pode conter valores NULL

Cada tabela deve ter uma chave primária e apenas uma chave primária

FOREIGN KEY

Uma FOREIGN KEY (Chave Estrangeira) em uma tabela é um campo que aponta para uma chave primária em outra tabela

```
CONSTRAINT fk_ID_Autor FOREIGN KEY(ID_Autor) REFERENCES  
tb_Autores(ID_Autor);
```

Neste exemplo a chave primária está na tabela tb_Autores e uma chave estrangeira de nome ID_Autor foi criada na tabela atual, usando o nome fk_ID_Autor

CHECK

A constraint é utilizada para limitar uma faixa de valores que podem ser colocados em uma coluna

Se uma constraint CHECK for definida em uma única coluna ela permitirá apenas determinados valores para a coluna

Se a constraint CHECK for definida para a tabela, ela poderá limitar os valores em algumas colunas com base nos valores de outras colunas do registro

DEFAULT

A restrição DEFAULT é usada para inserir um valor padrão em uma coluna

O valor padrão será adicionado a todos os novos registros caso nenhum outro valor seja especificado

03 – Normalização e Desnormalização

Quando criamos uma tabela em um banco de dados e incluímos colunas de outras tabelas, ou seja, referenciamos colunas chaveadas através de chaves estrangeiras utilizamos instruções por meio das constraints (restrições)

Com a criação das constraints de chave estrangeira, damos mais segurança à integridade dos dados. Se você tentar deletar algum registro da tabela que possui registro referenciado, o banco de dados não permitirá, mantendo a integridade

Vejamos um exemplo:

```
CREATE DATABASE db_Comercial  
ON PRIMARY(  
    NAME = db_Comercial,  
    FILENAME = 'C:\sql\db_Comercial.MDF',  
    SIZE = 6MB,  
    MAXSIZE = 15MB,  
    FILEGROWTH=10%  
);
```

Selecione o banco de dados com o comando USE nome_do_banco

Pense em uma modelagem de banco de dados que favorecerá as regras de negócio no que diz respeito a três tipos de entidades: Cliente, Pedido e Produto

A tabela Produto será criada com os seguintes campos (atributos):
ID_Produto, Descricao, Fabricante, Quantidade, Preco

A tabela Cliente será criada com os seguintes campos (atributos):
ID_Cliente, CPF, Nome, Salario

A tabela Pedido será criada com os seguintes campos (atributos):
ID_Pedido, Data, Quantidade, ID_Cliente, ID_Produto, Valor_Pedido

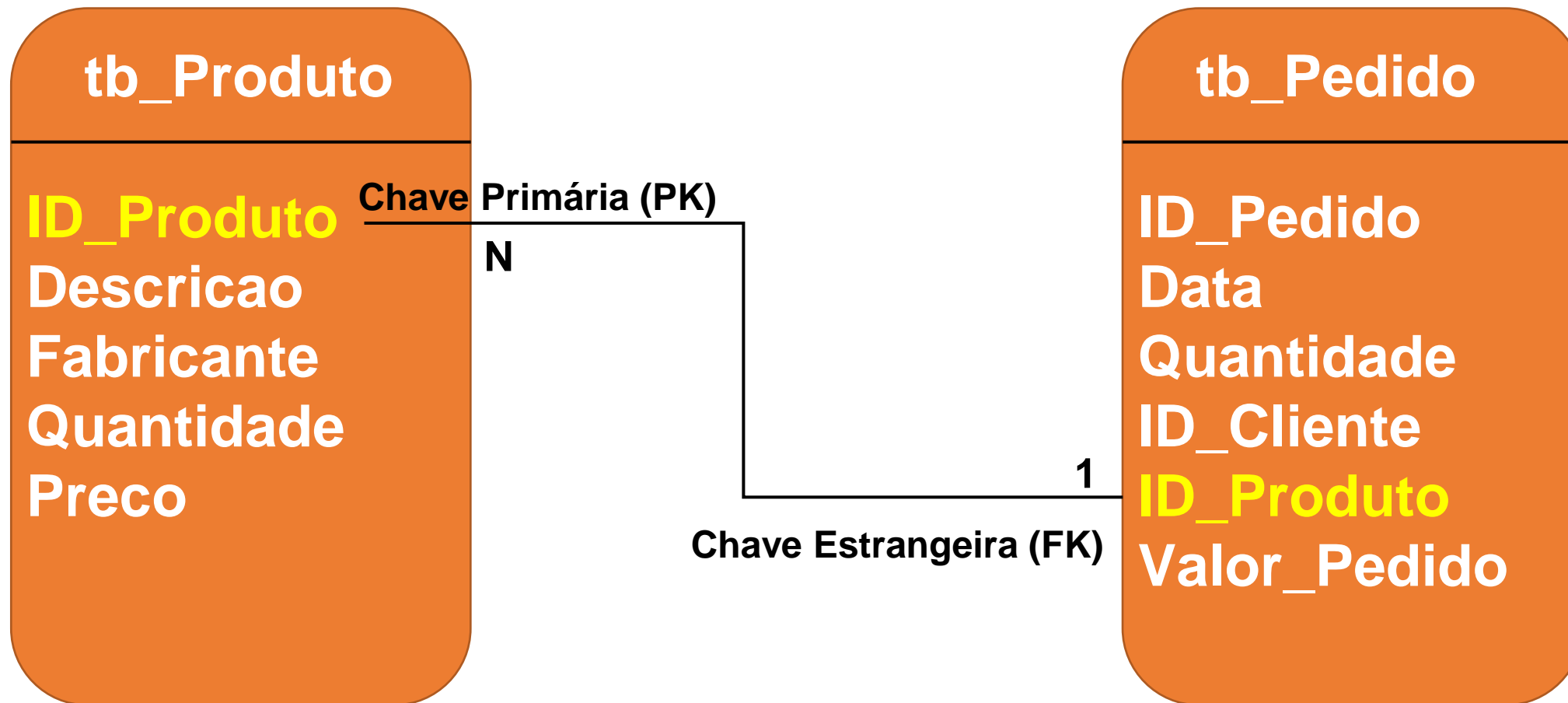
Para a normalização desse banco de dados, será necessário estabelecer relacionamentos entre as tabelas de forma que mantenham a integridade dos dados, dentro de uma regra de negócio

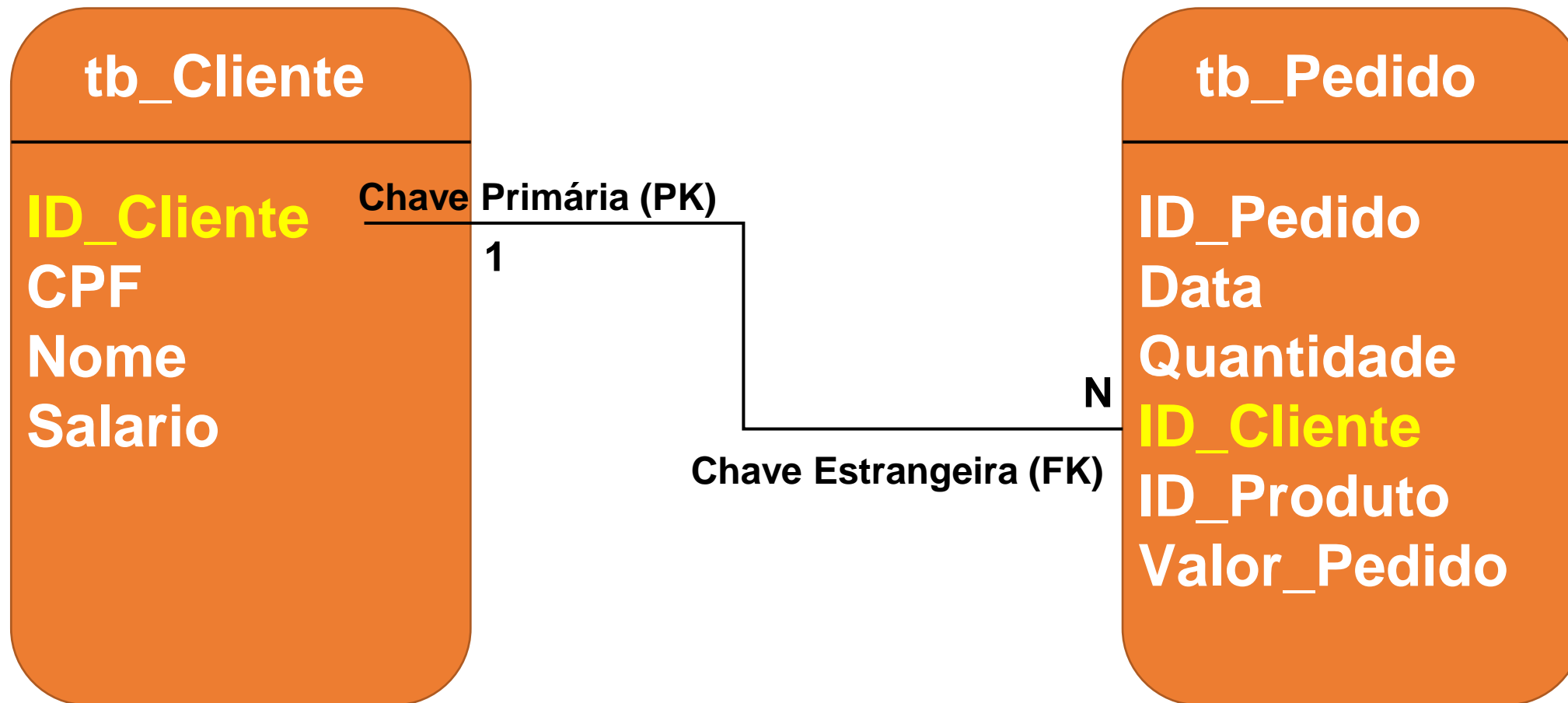
A tabela Pedido possui dois campos que se referem a chaves estrangeiras, e estes campos estão se referenciando a chave primária das respectivas tabelas:

ID_Cliente (foreign key) faz referência a ID_Cliente da tabela Cliente

ID_Produto (foreign key) faz referência a ID_Produto da tabela Produto

Dessa forma a integridade dos dados serão mantidas dentro dessa normalização





outra forma de normalizar nosso banco de dados nas tabelas é utilizando as constraints (**UNIQUE, NOT NULL, INDEX**), nesse exemplo, podemos definir os campos que são chaves estrangeiras como **NOT NULL** e o campos **CPF** como **UNIQUE**, ou seja, não podemos cadastrar clientes com CPF repetido

Dúvidas?

Professor Anderson Henrique



Para a próxima aula

01 – Criação de tabelas

02 – IDENTITY (Auto Increment)

03 – ALTER / DROP TABLE – Alterar e Excluir Tabelas e Colunas

Professor Anderson Henrique