



# Banco de Dados – Laboratório

Marco Montebello | FACENS  
[marco.montebello@facens.br](mailto:marco.montebello@facens.br)



# DDL – DATA DEFINITION LANGUAGE

LINGUAGEM DE DEFINIÇÃO DE DADOS. PARTE DO SQL UTILIZADA PARA DEFINIR DADOS E OBJETOS EM UM BANCO DE DADOS.

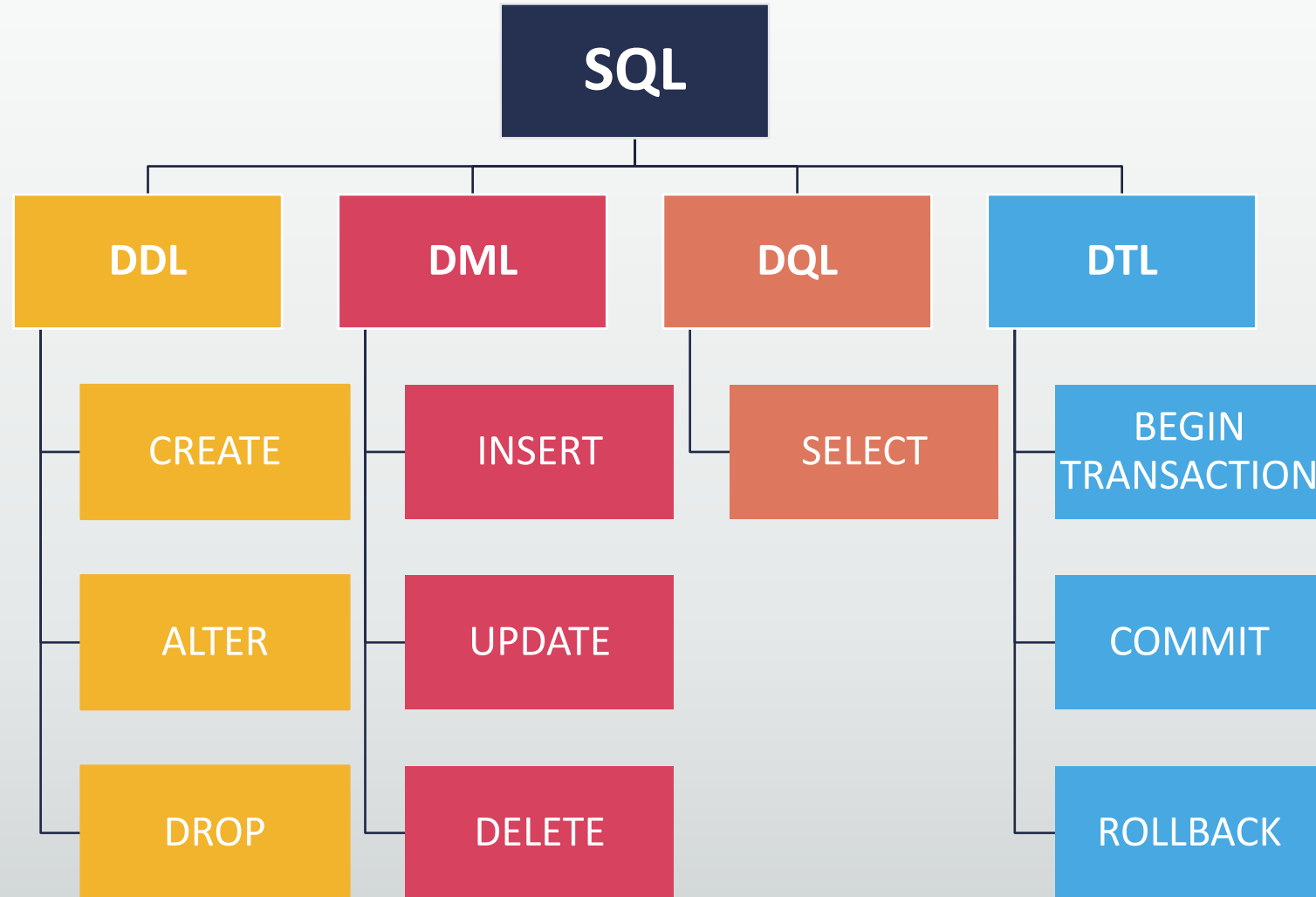


# Agenda

- Tipos de dados
- Restrições de Integridade
- Chaves Primária e Estrangeira
- Criação de Tabelas (*CREATE TABLE*)
- Alteração de Tabelas (*ALTER TABLE*)
- Exclusão de Tabelas (*DROP TABLE*)
- Demonstração
- Exercícios



# Definições Básicas





# DDL – *Data Definition Language*

- Linguagem de definição de dados (ou DDL, de *Data Definition Language*) é um conjunto de comandos dentro da SQL usada para a definição das estruturas de dados.
- Fornece instruções que permitem a criação, modificação e remoção das tabelas, assim como criação de índices.
- Estas instruções SQL permitem definir a estrutura de uma base de dados.
- Entre os principais comandos DDL estão **CREATE** (Criar), **ALTER** (Alterar) e **DROP** (Deletar).



# Tipos de Dados

Dados	Tipo
Caractere	char(n), varchar(n), nvarchar(n), nchar(n)
Numérico exato	decimal(p,e) ou numeric(p,e)
Numérico aproximado	float, real
Numérico inteiro	bigint, int, smallint, tinyint
Monetário	money, smallmoney
Data e hora	datetime, datetime2, smalldatetime, date, time
Binário	binary(n), varbinary(n)
Texto e imagens	text, image, ntext
Outros	bit, timestamp



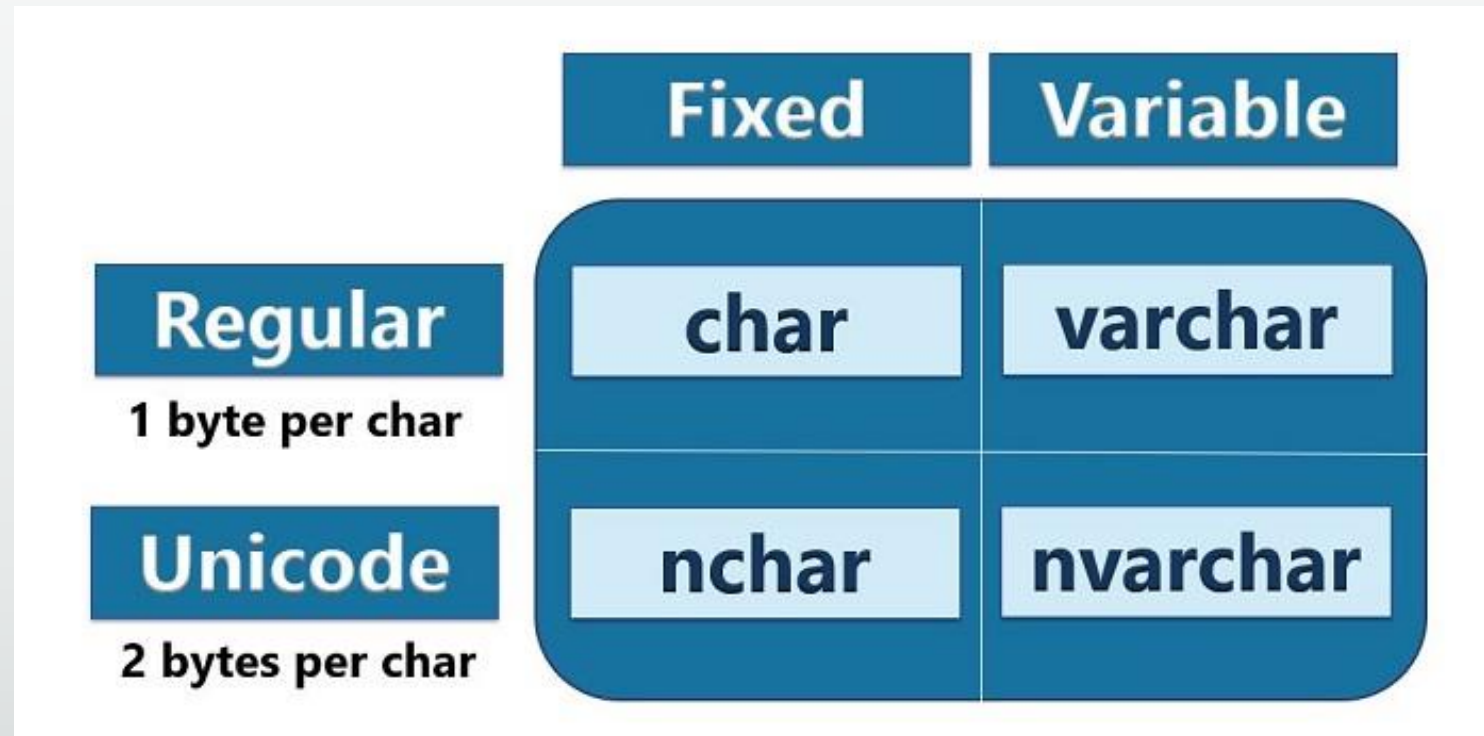
# Char *versus* Varchar

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
CHAR	Anna																				
	Pamela																				
	Bernardo																				
	Marco Montebello																				

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
VARCHAR	Anna																				
	Pamela																				
	Bernardo																				
	Marco Montebello																				



# Char *versus* Varchar





# Char *versus* Varchar

```
DECLARE @String1 char(10) = 'Hello'  
DECLARE @String2 varchar(10) = 'World'
```

```
SELECT @String1 + @String2  
SELECT LEN(@String1), LEN(@String2)  
SELECT DATALENGTH(@String1), DATALENGTH(@String2)
```

-----

```
DECLARE @String3 nchar(10) = N'Hello'  
DECLARE @String4 nvarchar(10) = N'World'
```

```
SELECT @String3 + @String4  
SELECT LEN(@String3), LEN(@String4)  
SELECT DATALENGTH(@String3), DATALENGTH(@String4)
```

# Restrições – *Constraints*

Restrição	Descrição
<b>Not Null</b>	Especifica que a coluna não pode conter um valores null.
<b>Primary Key</b>	Identifica unicamente cada linha da tabela.
<b>References ou Foreign Key</b>	Estabelece e força um relacionamento entre uma coluna e a tabela referenciada.
<b>Identity</b>	Especifica que o campo é auto incremento.
<b>Unique</b>	Especifica que uma coluna ou combinação de colunas devem ser únicas para todas as linhas na tabela.
<b>Check</b>	Especifica que uma condição deve ser <b><i>True</i></b> .
<b>Default</b>	Especifica um valor padrão.

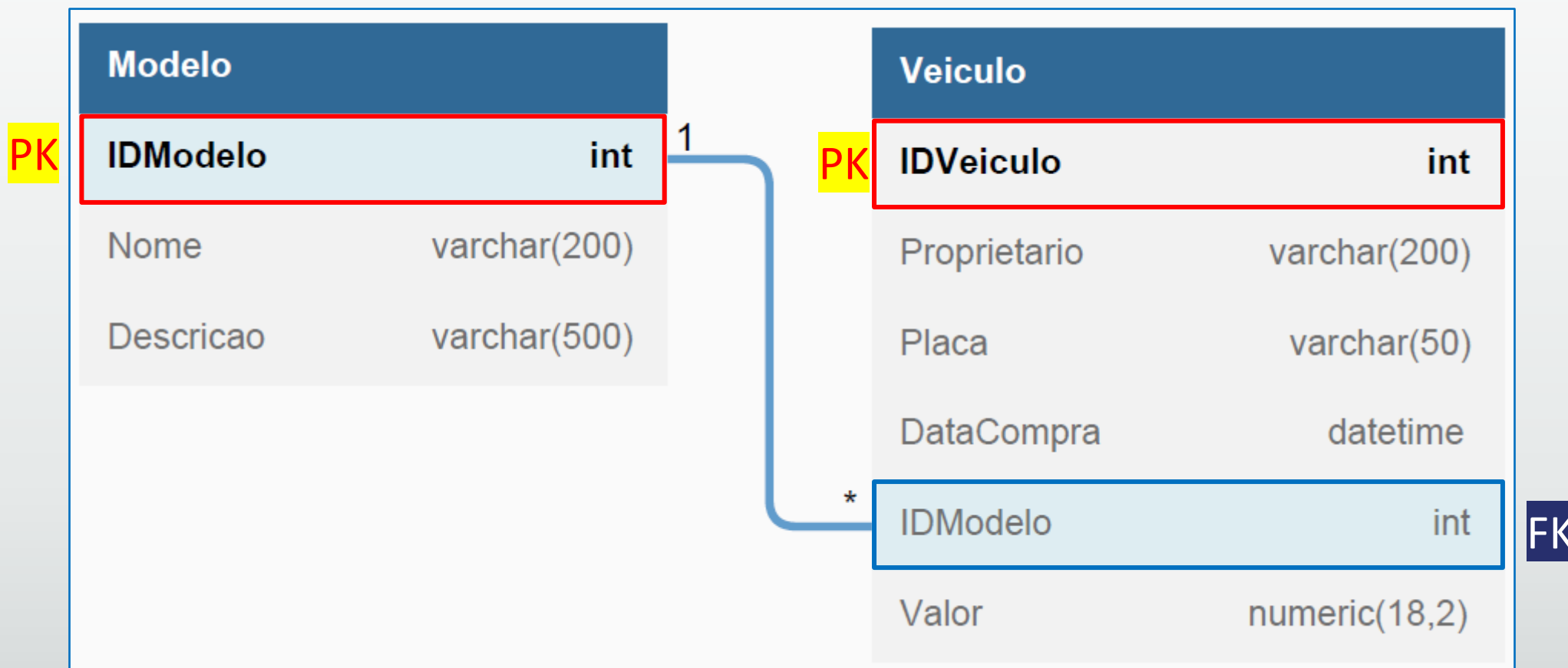


# Data Definition Language – DDL

- Os comandos DDL são utilizados para definir tabelas, suas regras e relacionamentos.
  - **CREATE TABLE**
  - **ALTER TABLE**
  - **DROP TABLE**



# Definição Chave Primária e Chave Estrangeira



# Chave Primária (*Primary Key*)

- Conceito mais básico relacionado à organização em um banco de dados.
- ***Toda tabela possuirá uma, e somente uma, chave primária.***
- Essa chave é utilizada como **identificador único da tabela**, sendo representada por aquele campo (ou campos) que não receberá valores repetidos.
  - Uma chave primária pode ser composta por um ou mais campos numa tabela. Quando são utilizados vários campos como chave primária, são denominados por chave composta
- Considerações:
  - Chaves primárias não podem ser nulas
  - Cada registro na tabela deve possuir uma, e somente uma, chave primária
  - Normalmente, chaves primárias são incrementadas automaticamente pelo banco de dados (***identity***)
  - São as chaves para o relacionamento entre entidades ou tabelas da base de dados. Assim haverá na tabela relacionada uma referência a essa chave primária (que será, na tabela relacionada, a chave estrangeira).



# Chave Estrangeira (*Foreign Key*)

- Está ligado ao relacionamento entre tabelas.
- É uma referência em uma tabela a uma chave primária de outra tabela.
- Observe o seguinte exemplo de duas tabelas: **Modelo** e **Veículo**.
- Para montarmos um relacionamento entre elas poderíamos ter na tabela **Veículo** o campo **IDModelo** fazendo referência à chave primária da tabela **Modelo**.
- Considerações
  - Pode ser nula (*NOT NULL*);
  - É um campo em uma tabela que faz referência a um campo que é chave primária em outra tabela
  - É possível ter mais de uma (ou nenhuma) em uma tabela

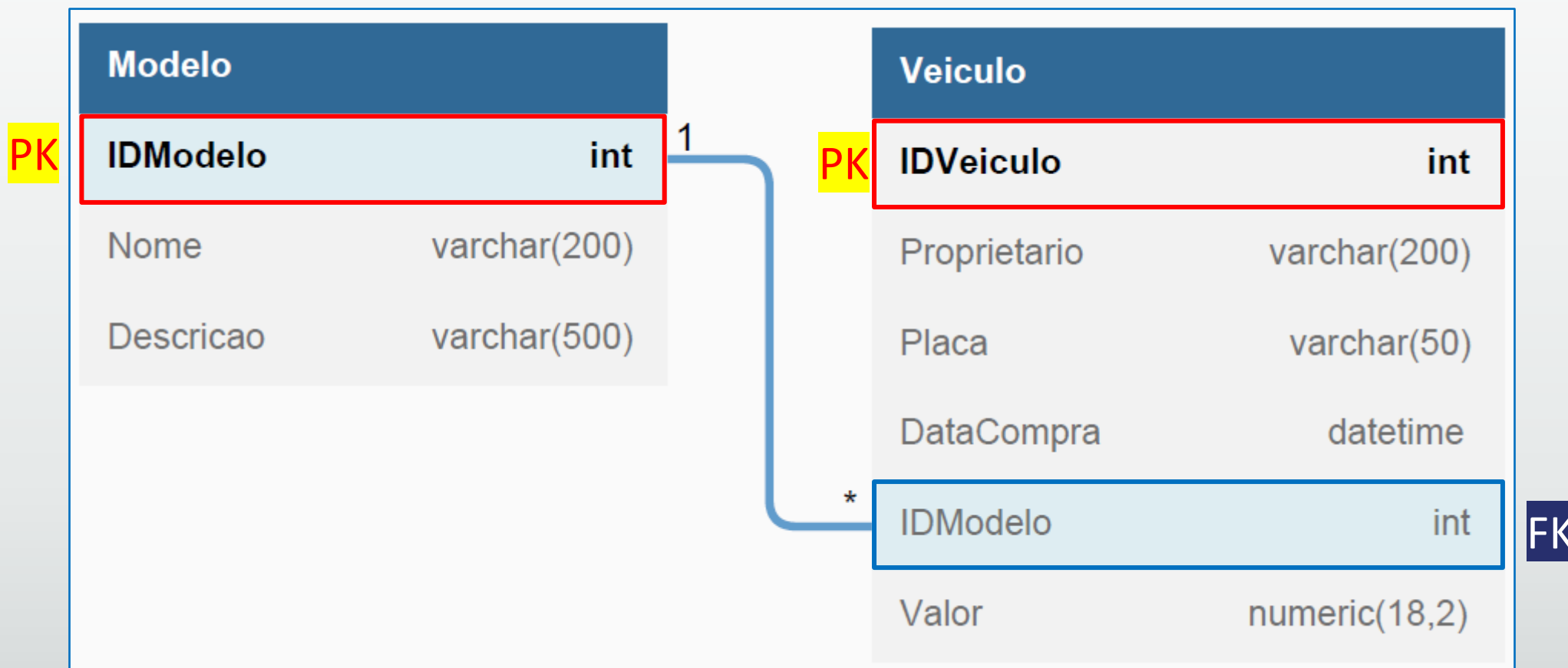




CREATE TABLE



# Definição Chave Primária e Chave Estrangeira



# Criação da Tabela: Modelo

## Sem nomear a Primary Key

```
CREATE TABLE Modelo
(
    IDModelo int not null PRIMARY KEY identity,
    Nome varchar(200) not null,
    Descricao varchar(500)
)
```

# Criação da Tabela: Veiculo

## Sem nomear a Foreign Key

```
CREATE TABLE Veiculo  
(  
    IDVeiculo int not null PRIMARY KEY identity,  
    Proprietario varchar(200) not null,  
    Placa varchar(50) not null,  
    DataCompra datetime,  
    IDModelo int not null REFERENCES Modelo(IDModelo),  
    Valor numeric(18,2) not null  
)
```



Chave  
Estrangeira

# Criação da Tabela: Modelo

## Nomeado a Primary Key (1)

```
CREATE TABLE Modelo
(
    IDModelo int not null identity,
    Nome varchar(200) not null,
    Descricao varchar(500),
    CONSTRAINT PKIDModelo PRIMARY KEY (IDModelo)
)
```

# Criação da Tabela: Veiculo

## Nomeado a Primary Key e Foreign Key (1)

```
CREATE TABLE Veiculo
(
    IDVeiculo int not null identity,
    Proprietario varchar(200) not null,
    Placa varchar(50) not null,
    DataCompra datetime,
    IDModelo int not null,
    Valor numeric(18,2) not null,
    CONSTRAINT PKIDVeiculo PRIMARY KEY (IDVeiculo),
    CONSTRAINT FKVeiculoModelo FOREIGN KEY (IDModelo) REFERENCES Modelo(IDModelo)
)
```



Chave Estrangeira



# Criação da Tabela: Modelo

## Nomeado a Primary Key (1)

```
CREATE TABLE Modelo
(
    IDModelo int not null identity,
    Nome varchar(200) not null,
    Descricao varchar(500)
)
```

```
ALTER TABLE Modelo ADD CONSTRAINT PKIDModelo PRIMARY KEY (IDModelo)
```

# Criação da Tabela: Veiculo

## Nomeado a Primary Key e Foreign Key (2)

```
CREATE TABLE Veiculo
```

```
(  
    IDVeiculo int not null identity,  
    Proprietario varchar(200) not null,  
    Placa varchar(50) not null,  
    DataCompra datetime,  
    IDModelo int not null,  
    Valor numeric(18,2) not null  
)
```

```
ALTER TABLE Veiculo ADD CONSTRAINT PKIDVeiculo PRIMARY KEY (IDVeiculo)
```

```
ALTER TABLE Veiculo ADD CONSTRAINT FKVeiculoModelo FOREIGN KEY (IDModelo) REFERENCES  
Modelo(IDModelo)
```



Chave  
Estrangeira

# Exemplo de Chave Primária Composta

```
CREATE TABLE ExemploChaveComposta  
(  
    IDChave1      int not null,  
    IDChave2      int not null,  
    Nome          varchar(150) not null  
)
```

```
ALTER TABLE ExemploChaveComposta ADD CONSTRAINT PKChaveComposta  
PRIMARY KEY(IDChave1, IDChave2)
```

# ALTER TABLE



# Alter Table – Adicionar Colunas

## ▪ Tabela Modelo

- Adicionar Colunas
  - NumeroLugares int
  - Tipo varchar(100)
- Adicionar Consistência
  - Tipo somente Sedan, SUV, Hatch e Outros
- Valor Padrão
  - NumeroLugares igual a 5

## ▪ Tabela Veiculo

- Adicionar Colunas
  - VouApagar numeric(18,2)
  - AnoFabricacao int
  - Cores varchar(10)
- Adicionar Consistência
  - AnoFabricacao como not null
- Tamanho da Coluna
  - Cores para varchar(50)
- Excluir coluna
  - VouApagar
- Renomear Coluna
  - Cores para Cor



# Alter Table – Modelo

--Adicionar Colunas

```
ALTER TABLE Modelo ADD NumeroLugares int
```

```
ALTER TABLE Modelo ADD Tipo varchar(100)
```

--Adicionar Consistência

```
ALTER TABLE Modelo ADD CHECK (Tipo in ('Sedan', 'SUV', 'Hatch', 'Outros'))
```

--Valor Padrão

```
ALTER TABLE Modelo ADD CONSTRAINT defLugares DEFAULT 5 FOR NumeroLugares
```



# Alter Table – Veiculo

## --Adicionar Colunas

```
ALTER TABLE Veiculo ADD VouApagar numeric(18,2)
```

```
ALTER TABLE Veiculo ADD AnoFabricacao int
```

```
ALTER TABLE Veiculo ADD Cores varchar(10)
```

## --Adicionar Consistência

```
ALTER TABLE Veiculo ALTER COLUMN AnoFabricacao int not null
```

## --Tamanho da Coluna

```
ALTER TABLE Veiculo ALTER COLUMN Cores varchar(50)
```

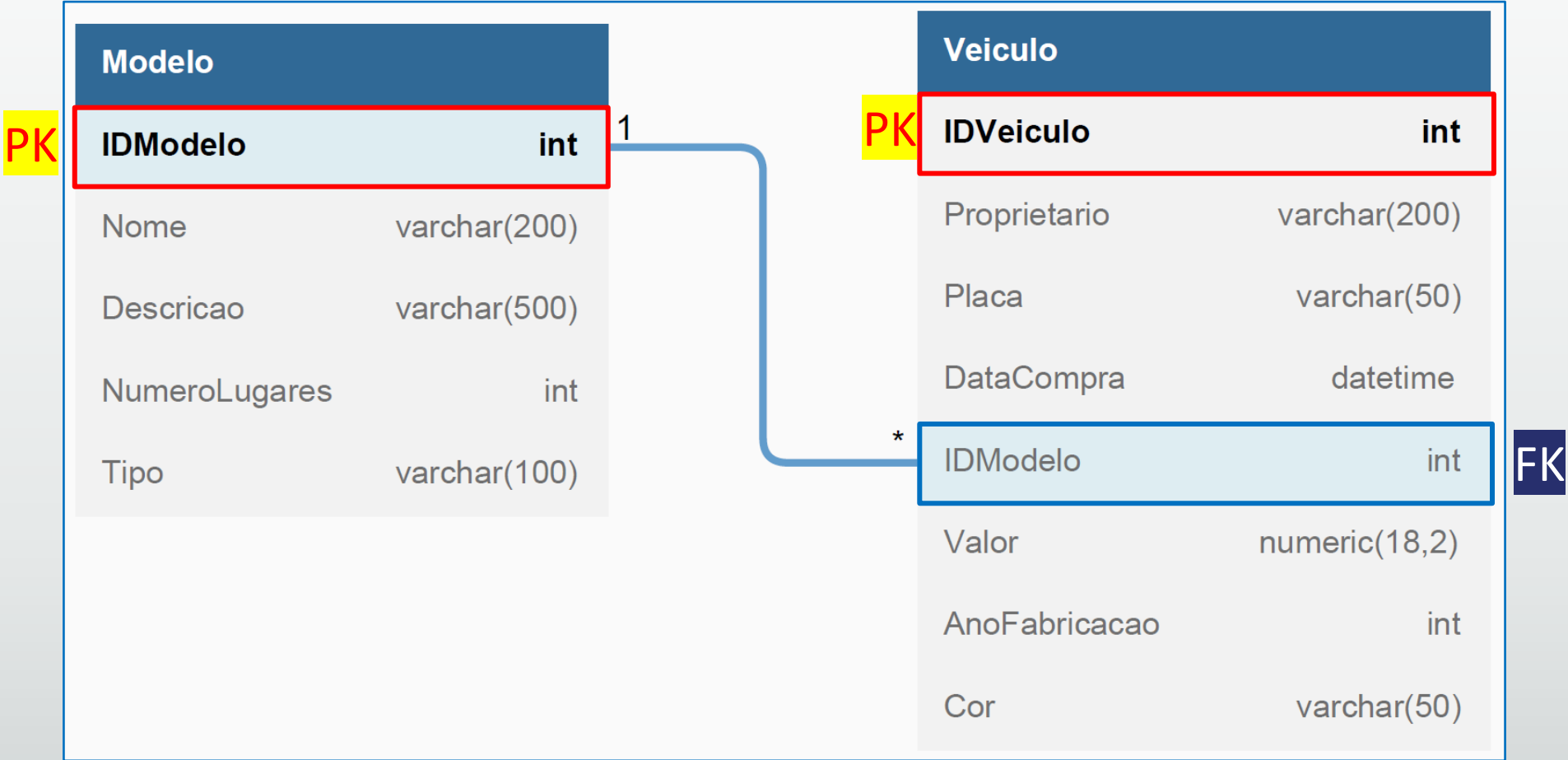
## --Excluir coluna

```
ALTER TABLE Veiculo DROP COLUMN VouApagar
```

## --Renomear Coluna

```
SP_RENAME 'Veiculo.Cores', Cor
```

# Modelo Após Alterações



# DROP TABLE



# Drop Table – Exemplos

```
DROP TABLE Veiculo
```

```
DROP TABLE Modelo
```



# Dúvidas???

marco.montebello@facens.br  
@montebellojr



# EXERCÍCIOS





A blurred office scene in the background. A woman in a black blazer is smiling and giving a high-five to a man in a grey shirt and dark vest. Another woman is visible in the background, working at a desk. The overall tone is professional and celebratory.

# Obrigado!!!

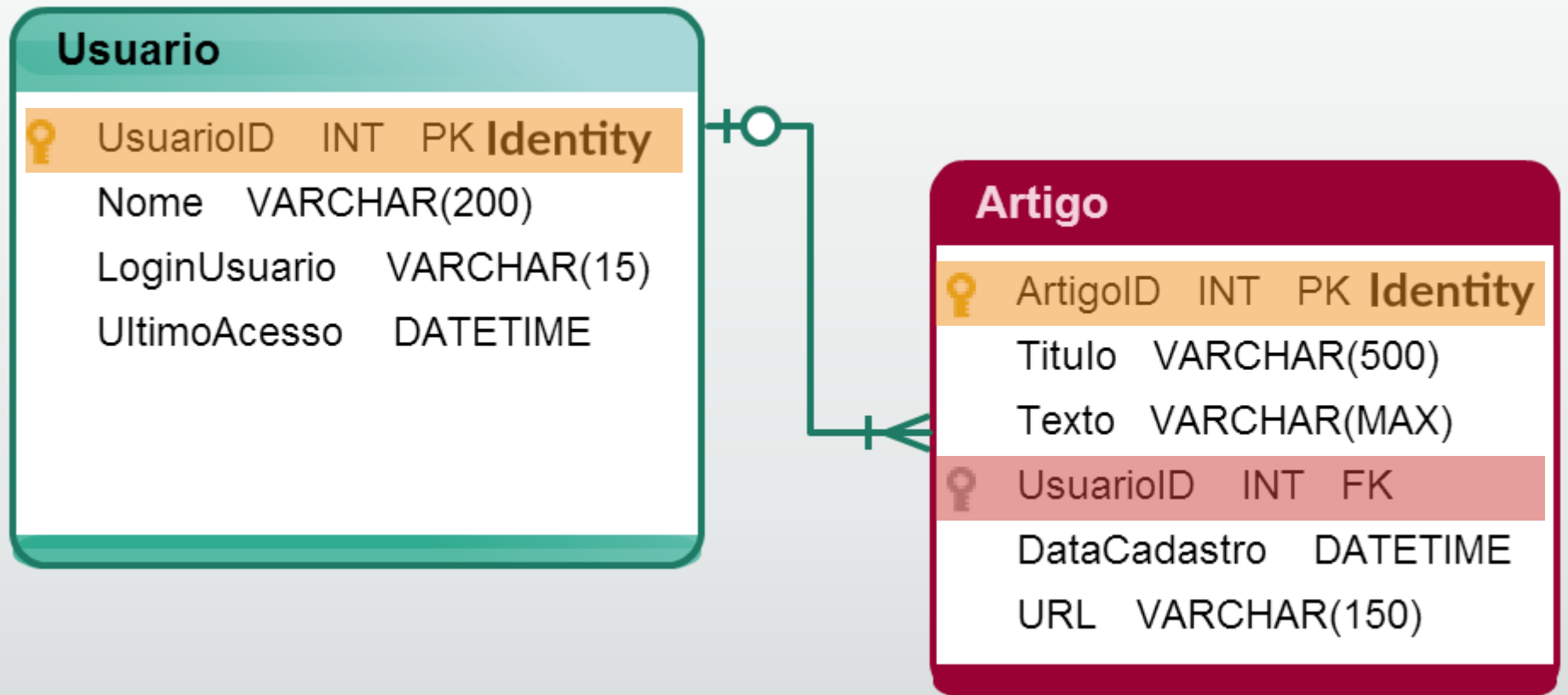
marco.montebello@facens.br  
@montebellojr



# OUTROS EXEMPLOS



# Create Table – Exemplo



[online diagramming & design] [creately.com](https://creately.com)



# Criação da Tabela: Usuário

## Sem nomear a Primary Key

```
CREATE TABLE Usuario
(
    UsuarioID      int not null identity PRIMARY KEY,
    Nome           varchar(200) not null,
    LoginUsuario   varchar(15) not null,
    UltimoAcesso   datetime
)
```

# Criação da Tabela: Artigo

## Sem nomear a Foreign Key

```
CREATE TABLE Artigo
(
    ArtigoID      int not null identity PRIMARY KEY,
    Titulo        varchar(500) not null,
    Texto         varchar(max),
    UsuarioID     int not null REFERENCES Usuario(UsuarioID),
    DataCadastro  datetime,
    URL           varchar(150)
)
```



Chave  
Estrangeira

# Criação da Tabela: Usuário

## Nomeado a Primary Key (1)

```
CREATE TABLE Usuario
(
    UsuarioID      int not null identity,
    Nome           varchar(200) not null,
    LoginUsuario   varchar(15) not null,
    UltimoAcesso   datetime,
    CONSTRAINT PKUsuarioID PRIMARY KEY(UsuarioID)
)
```

# Criação da Tabela: Artigo

## Nomeado a Primary Key e Foreign Key (1)

```
CREATE TABLE Artigo
```

```
(
```

```
    ArtigoID      int not null identity,
```

```
    Titulo        varchar(500) not null,
```

```
    Texto         varchar(max),
```

```
    UsuarioID     int not null,
```

```
    DataCadastro  datetime,
```

```
    URL           varchar(150),
```

```
    CONSTRAINT PKArtigoID PRIMARY KEY (ArtigoID),
```

```
    CONSTRAINT FKArtigoUsuario FOREIGN KEY (UsuarioID) REFERENCES Usuario(UsuarioID)
```

```
)
```



Chave  
Estrangeira

# Criação da Tabela: Usuário

## Nomeado a Primary Key (1)

```
CREATE TABLE Usuario
```

```
(
```

```
    UsuarioID      int not null identity,
```

```
    Nome           varchar(200) not null,
```

```
    LoginUsuario   varchar(15) not null,
```

```
    UltimoAcesso   datetime
```

```
)
```

```
ALTER TABLE Usuario ADD CONSTRAINT PKUsuarioID PRIMARY KEY(UsuarioID)
```



# Criação da Tabela: Artigo

## Nomeado a Primary Key e Foreign Key (2)

```
CREATE TABLE Artigo
(
    ArtigoID      int not null identity,
    Titulo        varchar(500) not null,
    Texto         varchar(max),
    UsuarioID     int not null,
    DataCadastro  datetime,
    URL           varchar(150)
)
```

```
ALTER TABLE Artigo ADD CONSTRAINT PKArtigoID PRIMARY KEY (ArtigoID)
```

```
ALTER TABLE Artigo ADD CONSTRAINT FKArtigoUsuario FOREIGN KEY (UsuarioID) REFERENCES
Usuario(UsuarioID)
```



Chave Estrangeira

# Alter Table – Adicionar Colunas

```
ALTER TABLE Usuario ADD Idade int
```

```
ALTER TABLE Usuario ADD Sexo char(1)
```

```
ALTER TABLE Artigo ADD Site varchar(100)
```

```
ALTER TABLE Artigo ADD VouApagar numeric(18,2)
```



# Alter Table – Adicionar Consistência

- Tabela Usuário
  - Sexo somente F ou M ou O
- Tabela Artigo
  - Site não pode ser nulo



# Alter Table – Adicionar Consistência

```
ALTER TABLE Usuario ADD CHECK (Sexo in ('F', 'M', 'O'))
```

```
ALTER TABLE Artigo ALTER COLUMN Site varchar(100) not null
```



# Alter Table – Tamanho da Coluna

- Tabela Artigo
  - Coluna Site varchar(1500)



# Alter Table – Excluindo uma Coluna

- Tabela Artigo
  - Coluna VouApagar



# Alter Table – Excluindo uma Coluna

```
ALTER TABLE Artigo DROP COLUMN VouApagar
```



# Alter Table – Renomear Colunas

- Tabela Artigo
  - Renomear a coluna Site para Hospedagem





# Alter Table – Resumo

- Adicionar Colunas
  - `ALTER TABLE Usuario ADD Idade int`
  - `ALTER TABLE Usuario ADD Sexo char(1)`
  - `ALTER TABLE Artigo ADD Site varchar(100)`
  - `ALTER TABLE Artigo ADD VouApagar numeric(18,2)`
- Adicionar Consistência
  - `ALTER TABLE Usuario ADD CHECK (Sexo in ('F','M','O'))`
  - `ALTER TABLE Artigo ALTER COLUMN Site not null`
- Tamanho da Coluna
  - `ALTER TABLE Artigo ALTER COLUMN Site varchar(1500)`
- Excluir coluna
  - `ALTER TABLE Artigo DROP COLUMN VouApagar`
- Renomear coluna
  - `SP_RENAME 'Artigo.Site', Hospedagem`
  - `SP_RENAME 'Artigo.URL', Link`



# Alter Table – Renomear Colunas

```
SP_RENAME 'Artigo.Site', Hospedagem
```

```
SP_RENAME 'Artigo.URL', Link
```

# Alter Table – Tamanho da Coluna

- Tamanho da Coluna
- `ALTER TABLE Artigo ALTER COLUMN Site varchar(1500)`





# OUTROS EXEMPLOS



# Create Table – Foreign Key

```
CREATE TABLE Consulta
```

```
(
```

```
    IDConsulta
```

```
    int PRIMARY KEY,
```

```
    Dataconsulta
```

```
    datetime,
```

```
    TipoConsulta
```

```
    char(1) CHECK (TipoConsulta in ('P','C')),
```

```
    IDPaciente
```

```
    int Not Null REFERENCES Paciente,
```

```
    IDMedico
```

```
    int Not Null REFERENCES Medico,
```

```
    ValorConsulta
```

```
    numeric(18,2) Not Null
```

```
)
```



Chave  
Estrangeira

# Create Table – Foreign Key

```
CREATE TABLE Consulta
```

```
(
```

```
    IDConsulta
```

```
    int PRIMARY KEY,
```

```
    Dataconsulta
```

```
    datetime,
```

```
    TipoConsulta
```

```
    char(1) CHECK (TipoConsulta in ('P','C')),
```

```
    IDPaciente
```

```
    int Not Null REFERENCES Paciente (IDPaciente),
```

```
    IDMedico
```

```
    int Not Null REFERENCES Medico (IDMedico),
```

```
    ValorConsulta
```

```
    numeric(18,2) Not Null
```

```
)
```



Chave  
Estrangeira

# Create Table – Foreign Key

```
CREATE TABLE Pedidos (  
    IDPedido          int PRIMARY KEY,  
    IDProduto         int REFERENCES Produtos (ProdutoID),  
    Quantidade        int  
)
```

Chave  
Estrangeira

```
CREATE TABLE Pedidos (  
    IDPedido          int PRIMARY KEY,  
    IDProduto         int REFERENCES Produtos,  
    Quantidade        int  
)
```

Chave  
Estrangeira



# Create Table – Valor Padrão

```
CREATE TABLE Exemplo (  
    ...  
    DataPagto      datetime DEFAULT (getdate()),  
    ...  
    Status         char(1)  DEFAULT 'F' CHECK (status in ('F','B')),  
    ...  
)
```

# Create Table – Valor Padrão

```
CREATE TABLE Produto (  
    IDProduto      int PRIMARY KEY,  
    Nome           varchar(100) ,  
    Preço          numeric(18,2) DEFAULT 9.99  
)
```

# Alter Table – Adicionar Colunas

```
ALTER TABLE Paciente ADD CodPaciente varchar(15)
```

```
ALTER TABLE Paciente ADD desconto varchar(1)  
check (desconto in ('S', 'N'))
```

```
ALTER TABLE Produto ADD descricao varchar(500)
```

# Alter Table – Adicionar Consistência

```
ALTER TABLE Paciente ADD CHECK (Sexo in ('F','M'))
```

```
ALTER TABLE Produto ALTER COLUMN ProdutoID int NOT NULL
```

# Alter Table – Tamanho da Coluna

```
ALTER TABLE Paciente ALTER COLUMN cidade varchar(300)
```

# Alter Table – Excluir Coluna

```
ALTER TABLE Paciente DROP COLUMN cidade
```

# Alter Table – Renomear Colunas

```
SP_RENAME `cliente.Endereco` , EnderecoCliente
```

# CHAVE PRIMÁRIA E ESTRANGEIRA





# Restrições – Chave Primária


## Forma simples (sem nomear a restrição)

```
CREATE TABLE Medico
(
    MedicoID      int PRIMARY KEY,
    NomeMedico    varchar(30)
)
```

# Restrições – Chave Primária

## Nomeando a restrição

```
CREATE TABLE Medico
(
    MedicoID      int,
    NomeMedico    varchar(30),
    CONSTRAINT PKMedicoID PRIMARY KEY (MedicoID)
)
```




Nome  
Chave  
Primária

# Restrições – Chave Primária

## Criando uma restrição após a criação da tabela

```
CREATE TABLE Medico  
(  
    MedicoID      int not null,  
    NomeMedico    varchar(30)  
)
```



Nome  
Chave  
Primária

```
ALTER TABLE Medico ADD CONSTRAINT PKMedicoID PRIMARY KEY (MedicoID)
```

# Restrições – Chave Estrangeira

## Forma mais simples (sem nomear a restrição)

```
CREATE TABLE Consulta
(
    ConsultaID          int PRIMARY KEY,
    DataConsulta        datetime,
    PacienteID          int REFERENCES Paciente (PacienteID),
    MedicoID             int REFERENCES Medico (MedicoID),
    Valconsulta         int
)
```

# Restrições – Chave Estrangeira

## Nomeando a restrição

```
CREATE TABLE Consulta
(
    ConsultaID          int PRIMARY KEY,
    DataConsulta        datetime,
    PacienteID          int,
    MedicoID            int,
    ValConsulta         numeric(5,2),
    CONSTRAINT FKConsultaPaciente FOREIGN KEY (PacienteID)
        REFERENCES Paciente,
    CONSTRAINT FKConsultaMedico FOREIGN KEY (MedicoID)
        REFERENCES Medico
)
```



Nome  
Chave  
Estrangeira

# Restrições – Chave Estrangeira

## Nomeando a restrição

```
CREATE TABLE Consulta
(
    ConsultaID          int PRIMARY KEY,
    DataConsulta        datetime,
    PacienteID          int,
    MedicoID            int,
    ValorConsulta       numeric(5,2),
)
```



Nome  
Chave  
Estrangeira

```
ALTER TABLE Consulta ADD CONSTRAINT FkConsultaPaciente FOREIGN KEY
(PacienteID) REFERENCES Paciente
```

```
ALTER TABLE Consulta ADD CONSTRAINT FkConsultaMedico FOREIGN KEY
(MedicoID) REFERENCES Medico
```