

Fundamentos de Banco de Dados

# Aula 7

# Tipos de Dados

- o Principais tipos de dados utilizados no SQL Server
- o **INT**: Valores numéricos inteiros variando de -2.147.483.648 até 2.147.483.647
- o **TINYINT**: Valores numéricos inteiros variando de 0 até 256
- o **SMALLINT**: Valores numéricos inteiros variando de -32.768 até 32.767
- o **BIGINT**: Valores numéricos inteiros variando de -92.23.372.036.854.775.808 até 9.223.372.036.854.775.807



# Tipos de Dados

- **BIT:** Somente pode assumir os valores 0 ou 1. Utilizado para armazenar valores lógicos.
- **DECIMAL(I,D) e NUMERIC(I,D):** Armazenam valores numéricos inteiros com casas decimais utilizando precisão. **I** deve ser substituído pela quantidade de dígitos total do número e **D** deve ser substituído pela quantidade de dígitos da parte decimal (após a vírgula). DECIMAL e NUMERIC possuem a mesma funcionalidade. Por exemplo, DECIMAL(8,2) armazena valores numéricos decimais variando de - 999999,99 até 999999,99
- Lembrando sempre que o SQL Server internamente armazena o separador decimal como ponto (.) e o separador de milhar como vírgula (,). Essas configurações INDEPENDEM de como o Windows está configurado no painel de controle e para DECIMAL E NUMERIC, somente o separador decimal (.) é armazenado

# Tipos de Dados

- **SMALLMONEY**: Valores numéricos decimais variando de -214.748,3648 até 214.748,3647
- **MONEY**: Valores numéricos decimais variando de -922.337.203.685.477,5808 até 922.337.203.685.477,5807
- **REAL**: Valores numéricos aproximados com precisão de ponto flutuante, indo de  $-3.40E + 38$  até  $3.40E + 38$
- **FLOAT**: Valores numéricos aproximados com precisão de ponto flutuante, indo de  $-1.79E + 308$  até  $1.79E + 308$



# Tipos de Dados

- o **SMALLDATETIME**: Armazena hora e data variando de 1 de janeiro de 1900 até 6 de junho de 2079. A precisão de hora é armazenada até os segundos.
- o **DATETIME**: Armazena hora e data variando de 1 de janeiro de 1753 até 31 de Dezembro de 9999. A precisão de hora é armazenada até os centésimos de segundos.
- o **CHAR(N)**: Armazena N caracteres fixos (até 8.000) no formato não Unicode. Se a quantidade de caracteres armazenada no campo for menor que o tamanho total especificado em N, o resto do campo é preenchido com espaços em branco.
- o **VARCHAR(N)**: Armazena N caracteres (até 8.000) no formato não Unicode. Se a quantidade de caracteres armazenada no campo for menor que o tamanho total especificado em N, o resto do campo não é preenchido.

# Tipos de Dados

- **TEXT:** Armazena caracteres (até 2.147.483.647) no formato não Unicode. Se a quantidade de caracteres armazenada no campo for menor que 2.147.483.647, o resto do campo não é preenchido. Procure não utilizar este tipo de dado diretamente, pois existem funções específicas para trabalhar com este tipo de dado.
- **NCHAR(N):** Armazena N caracteres fixos (até 4.000) no formato Unicode. Se a quantidade de caracteres armazenada no campo for menor que o tamanho total especificado em N, o resto do campo é preenchido com espaços em branco.



# Tipos de Dados

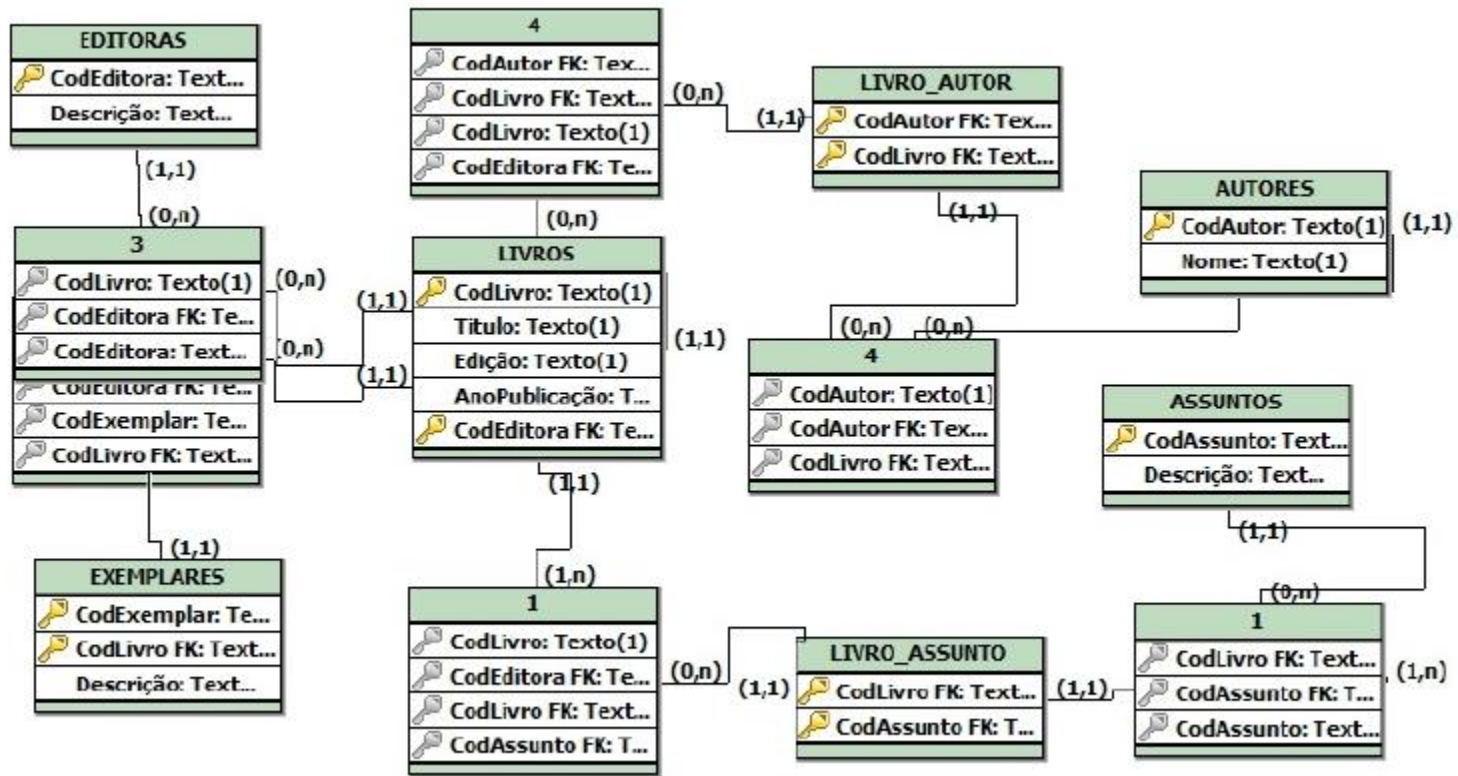
- o **NVARCHAR(N)**: Armazena N caracteres (até 4.000) no formato Unicode. Se a quantidade de caracteres armazenada no campo for menor que o tamanho total especificado em N, o resto do campo não é preenchido.
- o **NTEXT**: Armazena caracteres (até 1.073.741.823) no formato Unicode. Se a quantidade de caracteres armazenada no campo for menor que 1.073.741.823, o resto do campo não é preenchido. Procure não utilizar este tipo de dado diretamente, pois existem funções específicas para trabalhar com este tipo de dado.

# Modelagem Lógica

- o Representação da modelagem conceitual em um modelo de BD
- o Ênfase na eficiência de armazenamento
- o Preparação da implementação física



# Modelagem de Dados



# Modelagem Lógica

- o Na maioria dos programas de modelo ER podemos converter o modelo Conceitual em lógico.
- o É o caso do brModelo.



# Modelagem Lógica

- o O modelo lógico faz a representação mais próxima do modelo físico que será implementado.
- o Ele ainda é livre de SGBD, isto é, pode ser usado em qualquer SGBD.

# Modelagem Lógica

- o O modelo lógico tem ênfase em preparar as entidades em tabelas, atributos em colunas.
- o Nele é possível criar integridade de chaves, domínios de atributos.



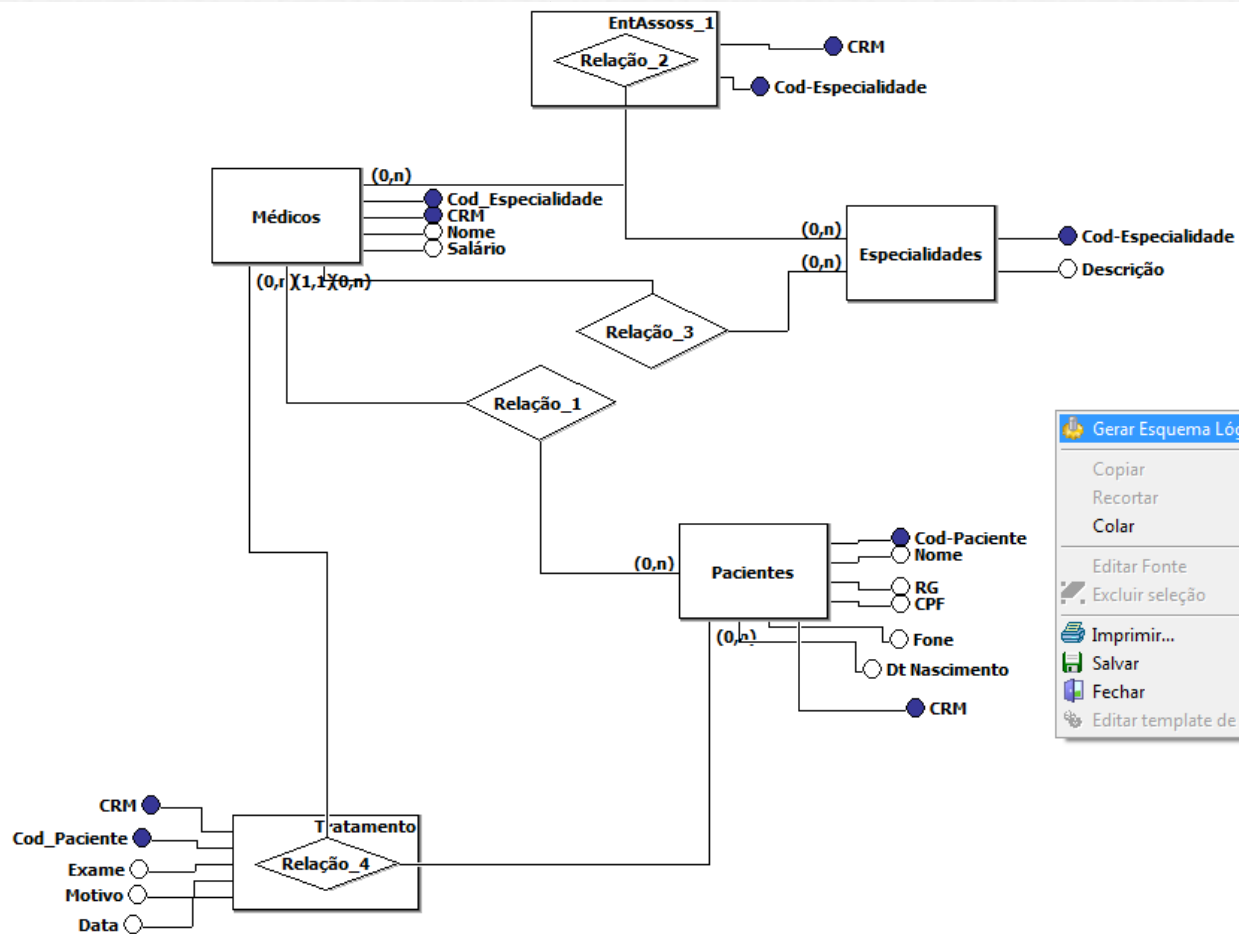
# Modelagem Lógica

- Para fazer a conversão do modelo conceitual em modelo lógico você deve primeiramente verificar se o modelo conceitual está concluído.
  - Cardinalidades mínimas e máximas definidas
  - Relacionamentos
  - Atributos
  - Entidades associativas

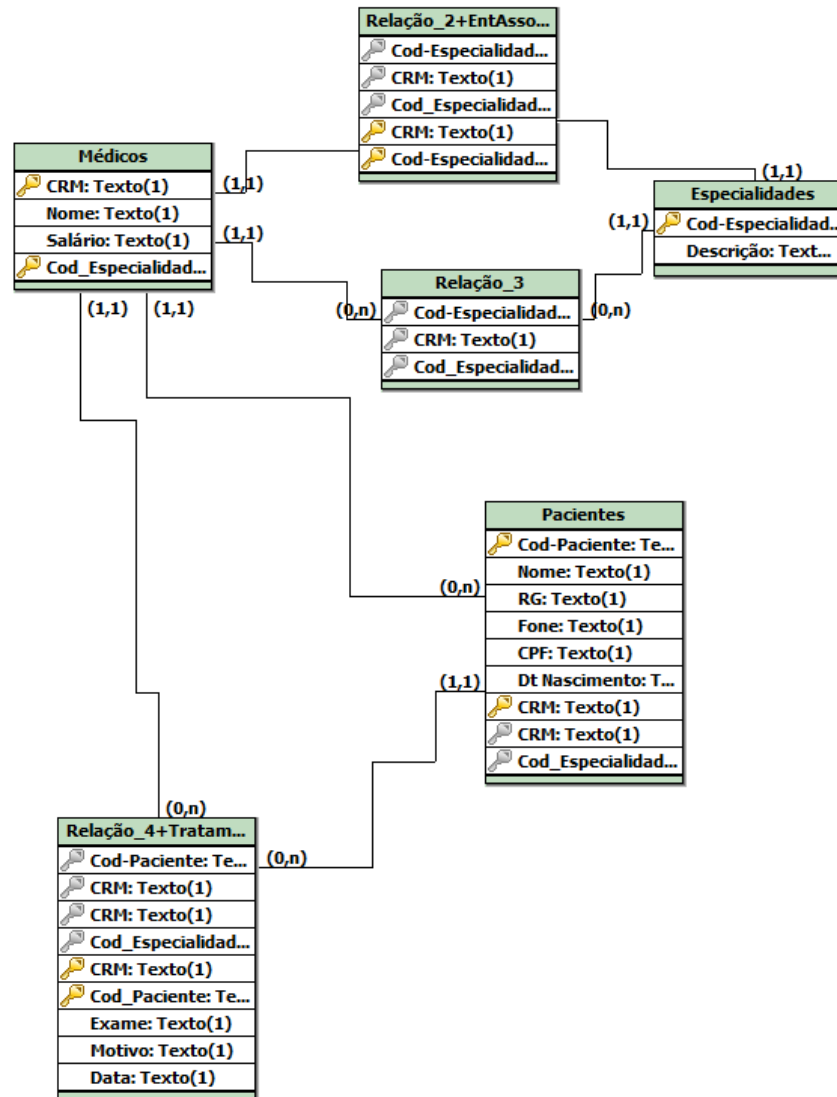
# Modelagem Lógica

- o No brModelo para criar seu modelo lógico, primeiramente é necessário ter o modelo Conceitual pronto.
- o No conceitual você deve ter criado todas as entidades, relacionamentos, entidades associativas necessárias e atributos.
- o Com ele pronto, clique com o direito em uma área vazia e clique em gerar esquema lógico





Gerar Esquema Lógico	
Copiar	Ctrl+C
Recortar	Ctrl+X
Colar	Ctrl+V
Editar Fonte	Ctrl+F
Excluir seleção	
Imprimir...	
Salvar	
Fechar	
Editar template de conversão	





# Modelagem Lógica

- o Após gerado o esquema lógico, deve-se prepará-lo. Ele não é gerado pronto.
- o Para isso é necessário revisar:
  - o Chaves primárias
  - o Chaves estrangeiras
  - o Entidades
  - o Relacionamentos
  - o Cardinalidades
  - o Entidades associativas (neste caso nota-se que a entidade associativa virou uma tabela)

# Modelagem Lógica

- o É necessário indicar o domínio do atributo, e indicar quando é chave, no caso de chave estrangeira, necessita indicar de onde vem esta chave, criando assim a integridade de chaves.
- o Os domínios de atributos, ficam a escolha de quem está criando, no nosso caso usaremos:
  - o Int
  - o Float
  - o Varchar(N)
  - o Datetime



# Modelagem Lógica

Seleção	Atr. ocultos
<b>Edição: Campo</b>	
Nome	CRM
Posição (Índice)	01
<b>Esquema</b>	
Chave Primária	Sim
Chave Estrangeira	Não
Tipo (Obrigatório):	Int
<b>IR</b>	
Tab. Origem	<nenhum>
Campo Origem	<nenhum>
<b>DDL</b>	
On Update	NO ACTION
On Delete	NO ACTION
Complemento	

Seleção	Atr. ocultos
<b>Edição: Campo</b>	
Nome	CRM
Posição (Índice)	07
<b>Esquema</b>	
Chave Primária	Não
Chave Estrangeira	Sim
Tipo (Obrigatório):	Int
<b>IR</b>	
Tab. Origem	Médicos
Campo Origem	CRM
<b>DDL</b>	
On Update	NO ACTION
On Delete	NO ACTION
Complemento	

Seleção	Atr. ocultos
<b>Edição: Campo</b>	
Nome	CRM
Posição (Índice)	01
<b>Esquema</b>	
Chave Primária	Sim
Chave Estrangeira	Sim
Tipo (Obrigatório):	Int
<b>IR</b>	
Tab. Origem	Médicos
Campo Origem	CRM
<b>DDL</b>	
On Update	NO ACTION
On Delete	NO ACTION
Complemento	

# Modelagem Lógica

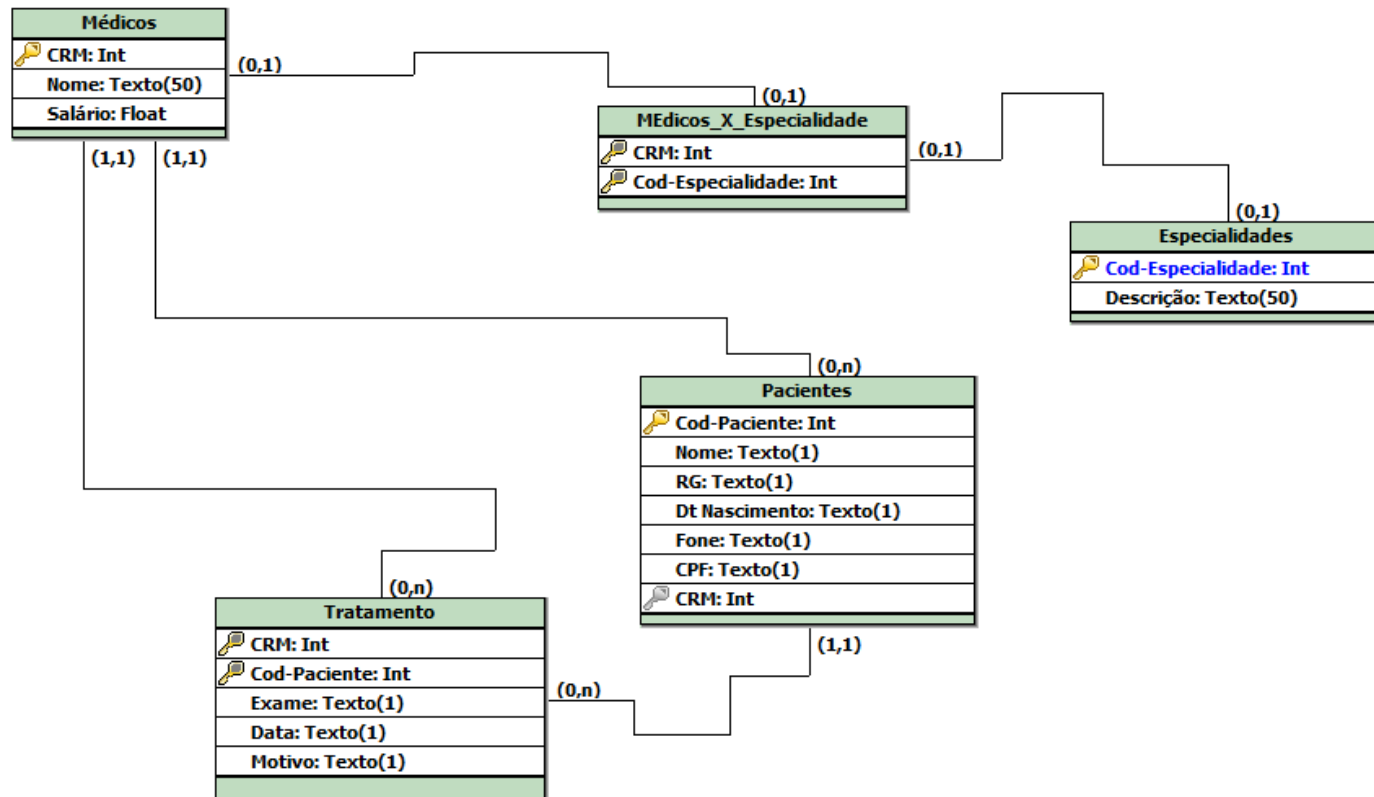
- o Verificando a necessidade do modelo lógico, pode-se incluir também novos atributos ou novos relacionamentos.
- o Usando a barra lateral da esquerda





# Modelagem Lógica

- Após as alterações e correções no modelo o resultado final poderá ser o seguinte:





# Modelagem de Dados

- o EXERCÍCIOS
- o No brModelo Implemente a modelagem Lógica para os modelos de negócio abaixo conforme criado no exercício da aula passada.

# Exercícios

## Seguros

- Construa um diagrama E-R para uma companhia de seguros de automóveis com um conjunto de clientes, onde cada um possui um certo número de carros.
- Cada carro tem um número de acidentes associados a ele.
- Cada acidente tem um número de carros associados.
- Defina os atributos que julgares necessário.



# Exercícios

## Administradora de Imóveis

- o Uma entrevista com o gerente da administradora resultou nas seguintes informações:
- o A administradora administra condomínios formados por unidades condominiais (lotes);
- o Cada lote pode pertencer a uma ou mais pessoas. Uma pessoa pode ter diversos lotes;
- o Cada lote pode estar alugado para no máximo uma pessoa. Uma pessoa pode alugar diversos lotes.
- o Defina os atributos que julgares necessário.