

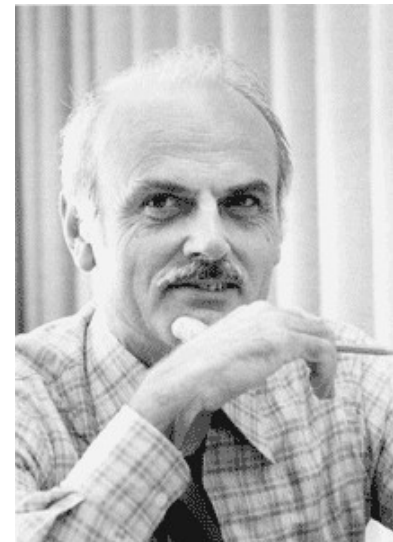
INSTITUTO FEDERAL  
PIAUÍ  
Campus Parnaíba

# Modelo Relacional

Prof. Msc Denival A. dos Santos

# Introdução

- Criado por **Edgar F. Codd** em 70 (primeiros sistemas relacionais: 1977 - 78: System R e Ingres);
- O **modelo relacional** é um modelo lógico fundamentado nos conceitos de uma relação matemática (teoria de conjuntos);
- A abordagem relacional está baseada no princípio de que as informações em uma base de dados podem ser consideradas como relações matemáticas e que estão representadas de maneira uniforme, através de tabelas bidimensionais;
- **Definição clássica:** São conjuntos de dados vistos segundo um conjunto de TABELAS e as operações sobre essas tabelas são feitas por linguagens que manipulam a álgebra relacional, manipulando conjuntos de uma só vez.



# Introdução

- O modelo relacional é hoje o modelo dominante para aplicações comerciais de banco de dados.
  - Oracle
  - SQL Server
  - DB2
  - Informix
  - Interbase
  - Firebird
  - MySQL
  - Etc.
- Levou ao desenvolvimento de uma poderosa linguagem de consulta, o SQL (Structured Query Language). Padrões ANSI SQL-86, ISO SQL-92 (SQL2), ISO SQL-99 (SQL3), SQL-2003, SQL-2006, SQL-2008, SQL-2011.

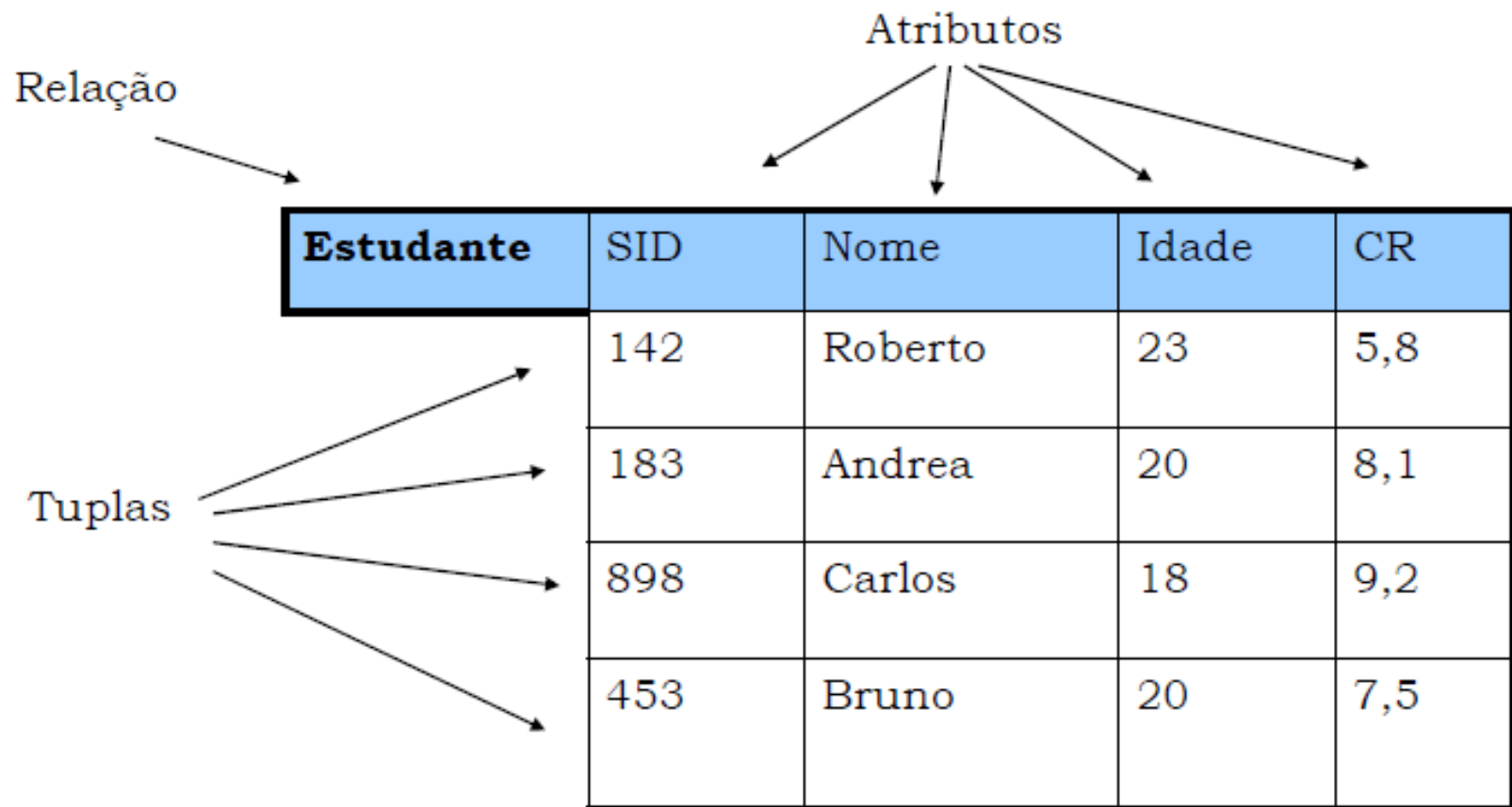
# Relação

- O modelo relacional é um modelo lógico onde os dados são representados em *relações* ou *tabelas*. Neste modelo, um banco de dados é um conjunto de relações;
- A terminologia tabela é mais comum nos produtos comerciais e na prática. Já a terminologia relação foi utilizada na literatura original sobre a abordagem relacional (daí a denominação relacional), sendo mais comum na área acadêmica e nos livros texto;
- Dá-se o nome de instância de uma relação ao conjunto de linhas de uma relação;
- Cada relação define uma tabela onde cada coluna é um atributo e cada linha uma tupla;
- Neste modelo, começa a preocupação em como os dados devem ser armazenados em como criar os relacionamento do modelo conceitual;
- Nesta etapa é definido o SGBD que será utilizado, bem como os tipos de dados para cada atributo.

# Terminologias do modelo

- **Domínio** - é uma coleção de valores atômicos, ou seja, valores “indivisíveis” dentro do contexto da situação a ser representada pelo banco de dados.
  - Exemplo: idade = inteiro; cep = conjunto de 8 caracteres, etc.
- **Atributo** - é um dado com nome e domínio definido, também chamado de campo da tabela. Também conhecido como coluna ou campo.
  - Exemplo: nome: string; idade: string; sexo [M,F], etc.
- **Grau** - número de atributos de uma relação.
- **Cardinalidade** - quantidade de tuplas da relação.
- **Tuplas** - conjunto de atributos com valores determinados. Também conhecida como linhas ou registros.
  - Não há tuplas duplicadas em uma mesma relação

# Terminologia do modelo - exemplo



# As 12 regras de Codd

- Codd, ao definir o modelo relacional, estabeleceu um conjunto de 12 regras para a determinação de um banco de dados ser realmente relacional. São elas:
  - 1 - Toda informação num banco de dados relacional é apresentada a nível lógico por valores em tabelas;
  - 2 - Todo dado em um banco de dados relacional tem a garantia de ser logicamente acessível, recorrendo-se a uma combinação do nome da tabela, um valor de chave e o nome da coluna;
  - 3 - Tratamento sistemático de valores nulos (ausência do dado);
  - 4 - O dicionário de dados (catálogo) relacional ativo é baseado no modelo relacional;
  - 5 - O SGBD relacional deve ter uma linguagem para definição, detalhamento e manipulação dos dados;

# As 12 regras de Codd

- 6 - Tratamento das atualizações de visões dos dados;
- 7 - Tratamento de alto nível para inserção, atualização e eliminação dos dados;
- 8 - Independência dos dados físicos (mudança na memória e no método de acesso);
- 9 - Independência de dados lógicos (mudanças de qualquer tipo nas tabelas básicas, ex: divisão de uma tabela por linha ou coluna);
- 10 - Independência das restrições de integridade;
- 11 - Independência de distribuição;
- 12 - Não subversão das regras de integridade ou restrições quando se utiliza uma linguagem de baixo nível.



# Chave

- Conceito básico para estabelecer relações entre linhas de tabelas de um banco de dados relacional é o da chave.
- Em um banco de dados relacional, há ao menos cinco tipos de chaves a considerar:
  - Chave candidata,
  - Chave primária,
  - Chave alternativa, e
  - Chave estrangeira.

# Chave candidata

- Uma **chave candidata** é uma superchave para qual nenhum subconjunto possa ser uma superchave.
- É possível que vários conjuntos diferentes de atributos possam servir como superchave, porém, nem todos serão chaves candidatas.
  - **Exemplo:** suponha uma combinação de nome\_cliente e rua\_cliente seja suficiente para distinguir todos os membros do conjunto de entidades cliente, assim como o atributo cpf, sozinho. Então (cpf) e (nome\_cliente, rua\_cliente) são chaves candidatas. Embora os atributos (cpf, nome\_cliente), juntos, possam, distinguir as entidades cliente, sua combinação não forma uma chave candidata, uma vez que cpf, sozinho, é uma chave candidata.
- É comum designar uma das chaves candidatas como chave primária da relação. As outras chaves candidatas são designadas como chave únicas (unique keys).

# Chave primária

- Uma **chave primária** é a chave candidata que é escolhida pelo projetista do banco de dados como de significado principal para a identificação de entidades dentro de um conjunto de entidades.
- A **chave primária** (ou *primary key* ou ainda **PK**) é um atributo da entidade que identifica apenas um objeto dessa entidade. Portanto, o valor dentro de uma chave primária não poderá se repetir e também não poderá receber um valor nulo.
  - **Exemplo:** Por exemplo, a entidade PROFESSOR tem tanto o atributo CPF quanto o atributo MATRICULA não se repetem, uma vez que esses atributos são únicos para cada indivíduo. Neste caso, qualquer um dos atributos poderia ser definido como uma chave primária. Então, qual escolher como chave primária? Solução: vai depender da modelagem, neste caso a matrícula seria a melhor escolha, uma vez que a maioria das consultas seriam feitas considerando a MATRICULA e não o CPF.

# Chave primária

- Um ponto importante a considerar durante a decisão de qual atributo deverá ser a chave primária é que se deve dar preferência a atributos numéricos (inteiros) em vez de atributos do tipo caractere, data ou hora.
- Uma chave primária pode ser:
  - **Simples** - é aquela que será formada por apenas um atributo.
    - Exemplo: matricula\_professor.
  - **Composta (ou concatenada)** - é composta por dois ou mais atributos.
    - Exemplo: numero\_compra, codigo\_produto.
- **Observação:** Não existe mais de uma chave primaria por relação. O que poderá existir é uma chave primária composta por mais de um atributo.

# Chave primária - exemplo

**TipoDeProduto:**

CodTipoProd	DescrTipoProd
1	Computador
2	Impressora

**Produto:**

CodProd	DescrProd	PrecoProd	QuantProd	CodTipoProd
1	PC desktop modelo X	2.500	5	1
2	PC notebook ABC	3.500	8	1
3	Impressora jato de tinta	600	3	2
4	Impressora laser	800	2	2

- Em um SGBD relacional, a chave primária é diferenciada dos demais atributos por possuir na sua representação textual (esquema) o atributo que será a chave primária da relação sublinhado (ou com um asterisco, ou com a sigla PK) .
- Exemplo:

**TipoDeProduto (CodTipoProd, DescrTipoProd)**

**Produto(CodProd, DescrProd, PrecoProd, QuantProd, CodTipoProd)**

# Chave estrangeira

- Um conceito muito importante quando se fala de modelo relacional é o conceito de **chave estrangeira** (ou *Foreign key* ou **FK**);
- Uma chave estrangeira é um atributo da tabela que faz referência a uma chave primária de outra tabela ou da própria tabela;

Alunos			
<u>Matrícula</u>	Nome	Endereço	<u>Turma</u>
200902003	Everson S. Araujo	Rua A, nº 12	SIG7
200902004	Heitor M. Araujo	Rua B, nº 34	SIG7
200903005	Vanusa M. Viana	Av. C, nº 19	GES3

Chave estrangeira

Turmas		
<u>Turma</u>	Curso	Período
SIG7	Sistema de Informação	7
GES3	Gestão de Negócios	3

# Chave estrangeira

- Exemplo:

Pai( CodPai (PK), nome)

Filho (CodFilho (PK), nome, CodPai (FK))

- Uma das notações utilizadas na literatura nos permite identificar mais facilmente a que chave primária de qual tabela a chave estrangeira faz referência. Nesta notação temos:

- Exemplo:

Pai( CodPai (PK), nome)

Filho (CodFilho (PK), nome, CodPai)

**CodPai referencia Pai**

- **Observação:** Uma chave estrangeira sempre faz referência a uma chave primária. A chave estrangeira nunca fará referência a um atributo que não seja uma chave primária. O nome do atributo que é chave estrangeira não precisa ser igual ao nome da chave primária correspondente.

# Restrições de integridade - RI

- Condição que é especificada sobre um esquema de relação e deve ser satisfeita para qualquer instância do BD.
  - RIs são especificadas na definição do esquema;
  - RIs são checadas pelo SGBD quando o BD é modificado.
- Objetivos
  - Evitar que o BD entre em um estado inconsistente
- RIs podem ser especificadas e forçadas em diferentes níveis.
  - Durante a criação de um esquema de relação
    - Exemplo: o domínio de um atributo.
  - Durante a execução de uma aplicação o SGBD impede operações que violem as RIs.



# Restrições de integridade - RI

- Um dos objetivos primordiais de um SGBD é a **integridade de dados**. Dizer que os dados de um banco de dados estão íntegros significa dizer que eles refletem corretamente a realidade representada pelo banco de dados e que são consistentes entre si. O controle de integridade de dados embutido nos SGBDs garante que todos os dados incluídos ou alterados no banco estarão de acordo com regras criadas para estes. Na abordagem relacional, costuma-se classificar as restrições de integridade nas seguintes categorias:
  - ← **Integridade de domínio** - restrições deste tipo especificam que o valor de um campo deve obedecer a definição de valores admitidos para a coluna (o domínio da coluna);
  - ← **Integridade de chave** - não podem existir duas tuplas de uma relação com valores iguais na chave primária;
  - ← **Integridade de vazio (entidade)** - a chave primária não pode conter um valor nulo (NULL). O NULL não é o valor 0 (zero) nem o caractere branco, é simplesmente a não existência de conteúdo neste campo;
  - ← **Integridade referencial** - é a restrição que define que os valores dos campos que aparecem em uma chave estrangeira devem aparecer na chave primária da tabela referenciada.

# Integridade Referencial com opção de exclusão

■ Notação:  $R_1[fk] \xrightarrow{op} R_2[pk]$

onde “op” é a opção de exclusão, dentro as seguintes:

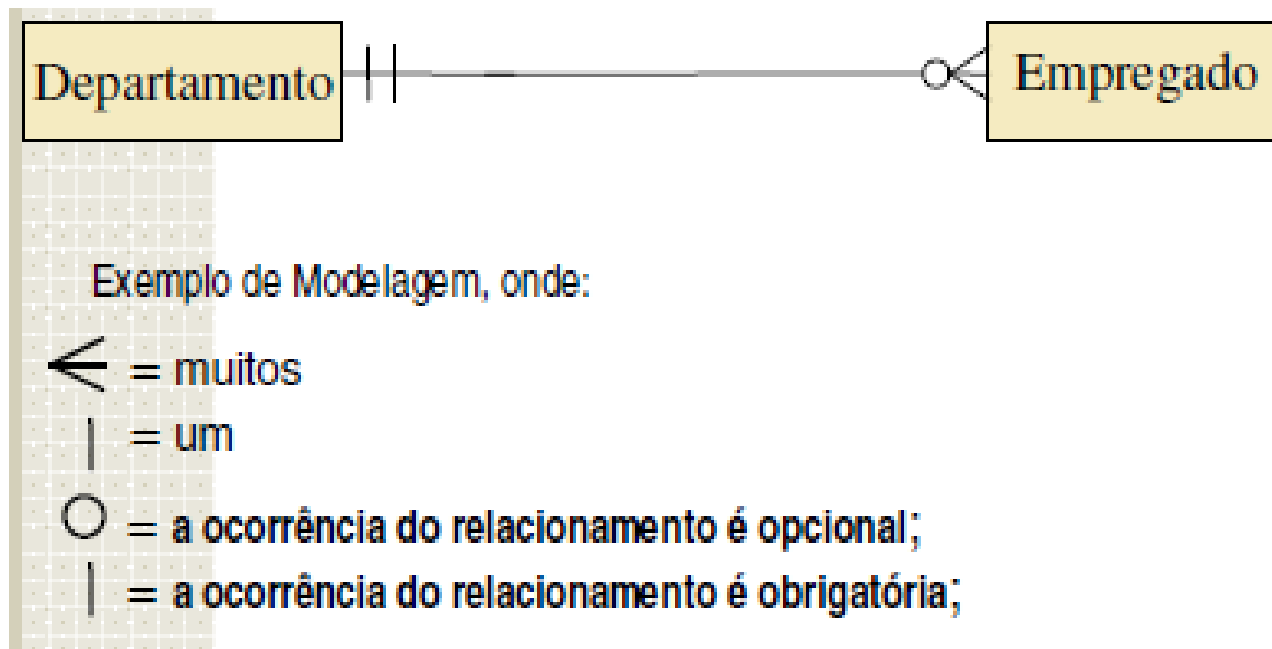
- **Bloqueio (restrict):** se alguma tupla referencia a tupla a ser excluída, através de uma chave estrangeira, a exclusão não é efetuada;
- **Propagação (cascade):** todas as tuplas que referenciam a tupla a ser excluída, através de uma chave estrangeira, são excluídas também automaticamente;
- **Substituição por nulos (set null):** todas as tuplas que referenciam a tupla a ser excluída, através de uma chave estrangeira, têm os valores dos atributos da chave estrangeira modificada para nulo (se for permitido nulo) e a exclusão é efetuada.

# Operações de atualização sobre relações

- **Inserção (insert)** - insere novas tuplas em uma relação.
  - Pode violar qualquer dos quatro tipos de restrições discutidas.
- **Exclusão (delete)** - exclui tuplas de uma relação
  - Pode violar somente restrições de integridade referencial.
- **Modificação (update ou modify)** - muda os valores de alguns atributos em tuplas existentes.
  - Modificar um atributo que não é chave primária nem chave estrangeira pode violar somente a restrição de domínio;
  - Modificar a chave primária é similar a excluir uma tupla e inserir uma outra no seu lugar; assim, pode violar qualquer das quatro restrições discutidas;
  - Modificar um atributo de uma chave estrangeira pode violar a restrição de integridade referencial ou de domínio.

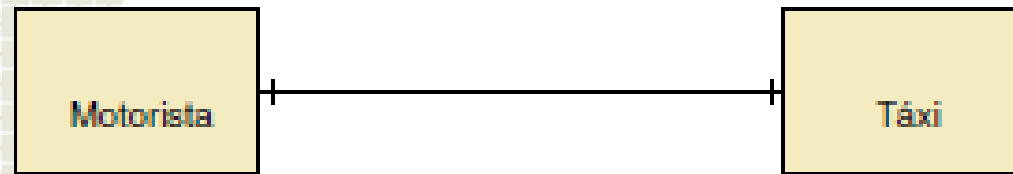
# Diagrama de tabelas relacionais - DTR

- O diagrama de tabelas relacionais é uma representação gráfica deste modelo, normalmente utilizamos a notação Crow's Foot de James Martin (comumente conhecida como “Pé de Galinha”, devido a simbologia utilizada). A maioria das ferramentas CASE atuais implementam apenas este diagrama.

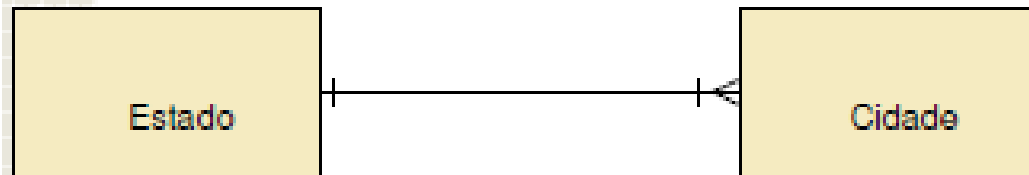


# Diagrama de tabelas relacionais - DTR

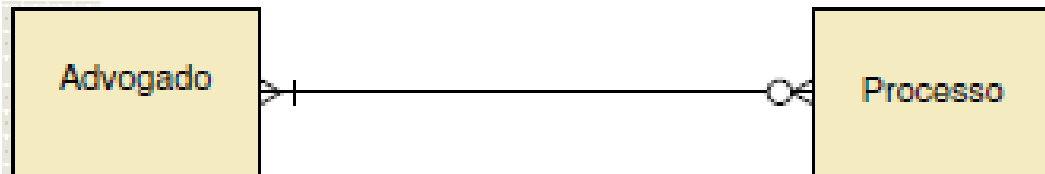
1:1



1:N

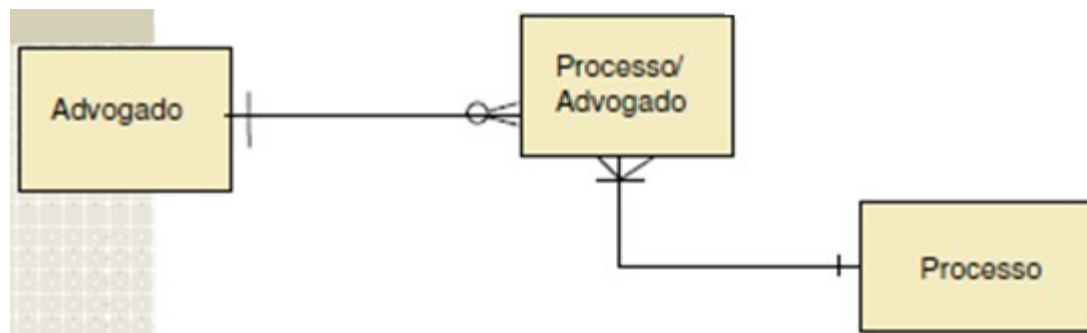


N:N



- **Observação:** Em BD relacionais não é possível implementar uma relacionamento N:M. Quando acontecer deve ser transformado em 1:N | M:1.

1:N | M:1



# Dicionário de dados

- Após o modelo relacional ter sido descrito ou diagramado, é necessário criar um Dicionário de dados para a base de dados.
- O dicionário de dados tem por objetivo descrever as propriedades de uma tabela, sua estrutura física e as restrições que cada atributo possui. Assim, o desenvolvedor irá implementar o banco de dados sabendo exatamente como a base deve ser criada;
- No dicionário de dados cada tabela da modelagem deverá ser descrita e deverá conter os seguintes campos: nome do atributo, descrição do atributo, tamanho, tipo, restrições (valor nulo, regra de domínio, chaves, valor default e unique).

# Dicionário de dados - exemplo

Nome	Descrição	Tipo	Tamanho	Nulo	Regra (check)	Chave	Default	Unique
matricula_aluno	Armazena a matricula do aluno	Numérico	5	Não	—	PK	—	Não
RG_aluno	Armazena o RG do aluno	Caracter	11	Não	—	—	—	Sim
nome_aluno	Armazena o nome do aluno	Caracter	100	Não	—	—	—	Não
data_nascimento_aluno	Armazena a data de nascimento do aluno	Data	—	Não	—	—	—	Não
cidade_aluno	Armazena a cidade em que o aluno mora	Caracter	20	Sim	—	—	Curitiba	Não
matricula_aluno_representante	Armazena a matricula do aluno representante	Numérico	5	Sim	—	—	—	Não
codigo_turma	Armazena o código da turma do aluno	Inteiro	—	Não	—	FK que referencia tbTurma	—	Não
sexo_aluno	Armazena o sexo a que o aluno pertence	Caracter	1	Não	M – Masculino F – Feminino	—	—	Não