

1

Como identificar atributos, entidades e keys



2

Como inserir informações nas tabelas



3

Como definir relacionamentos entre tabelas



4

Como montar o modelo no SQLBDM à partir da referência fornecida pelo professor

PASSO A PASSO ATIVIDADE DE MODELAGEM

+ Explicações de conceitos

SOBRE O MATERIAL

Aqui estão instruções de como pegar o modelo-referência que o professor passou e montá-lo no SQLBDM incluindo um resumo da matéria dada até agora com conceitos

Fontes: Uma caralhada de sites, um livro empoeirado, fóruns no stackoverflow e, obviamente, o chatGPT

PASSO 1

CONCEITOS



ENTIDADES

Em um banco de dados, as entidades são **objetos do mundo real ou conceitos abstratos que possuem características que precisam ser armazenadas** e gerenciadas. Por exemplo, em um sistema de gerenciamento de biblioteca, as entidades poderiam ser livros, autores e leitores.

EXEMPLO:

pessoa		
codigo_pessoa	int	PK
nome	varchar(50)	
telefone	varchar(50)	
celular	varchar(50)	
cpf	varchar(50)	
rg	varchar(50)	

No caso da biblioteca, para fazer o registro de uma pessoa, utilizaremos a entidade pessoa.

PASSO 1

CONCEITOS



ATRIBUTOS

Os atributos são as características ou propriedades de uma entidade que precisam ser armazenadas e gerenciadas. Por exemplo, para a entidade "livro", os atributos poderiam ser o título, o autor, a editora, o número de páginas e a data de publicação.

EXEMPLO:

pessoa		
codigo_pessoa	int	PK
nome	varchar(50)	
telefone	varchar(50)	
celular	varchar(50)	
cpf	varchar(50)	
rg	varchar(50)	

No caso da biblioteca, os atributos de uma pessoa representam suas características .

PASSO 1

CONCEITOS



CHAVE PRIMÁRIA

Uma chave primária é um atributo ou conjunto de atributos que identifica de forma única cada registro de uma tabela. Ela garante que não haverá duplicatas na tabela e é usada para estabelecer relacionamentos com outras tabelas.

Cada tabela deve ter uma chave primária.

EXEMPLO:

pessoa		
codigo_pessoa	int	PK
nome	varchar(50)	
telefone	varchar(50)	
celular	varchar(50)	
cpf	varchar(50)	
rg	varchar(50)	

No caso da biblioteca, a chave primária para uma pessoa é o "código_pessoa", uma sequência numérica que representa uma pessoa específica

PASSO 1

CONCEITOS



CHAVE COMPOSTA

A chave composta é uma chave primária que é formada por dois ou mais atributos, em vez de apenas um. Isso permite uma **identificação ainda mais precisa** dos registros em uma tabela, considerando diferentes características que juntas são únicas.

EXEMPLO:

No exercício da biblioteca não existem chaves compostas, no entanto caso tenha curiosidade, vários exemplos podem ser encontrados nessa thread do Stack Overflow

<https://pt.stackoverflow.com/questions/15883/quando-é-recomendado-o-uso-de-chave-primária-composta>

PASSO 1

CONCEITOS

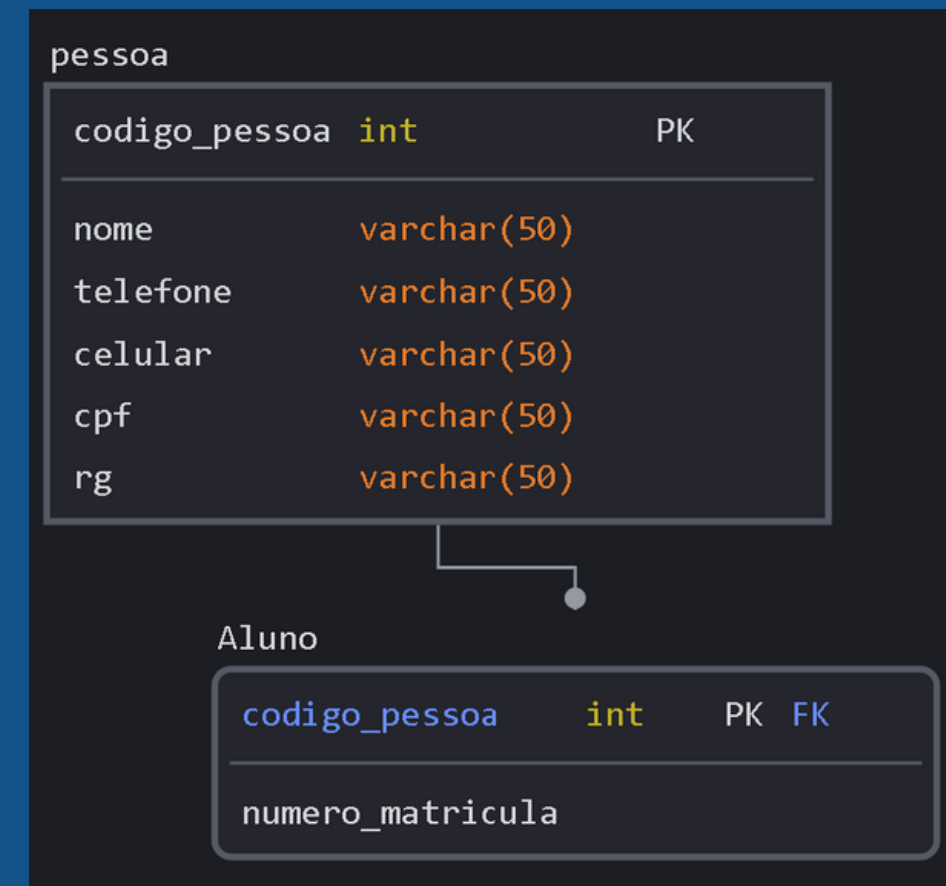


CHAVE ESTRANGEIRA

A chave estrangeira é um atributo de uma tabela que faz referência a uma chave primária em outra tabela, estabelecendo uma relação entre elas.

*Veja mais detalhes sobre relações no passo 3

EXEMPLO:



No caso da biblioteca, a chave "código_pessoa" é estrangeira por que foi importada da entidade "pessoa"

PASSO 1

CONCEITOS



TIPOS DE DADOS

O PostGres (sistema de banco de dados que usaremos para a atividade) suporta vários tipos de dados que podem ser acessados no link abaixo.

No entanto, para a atividade iremos usar apenas 3 deles

<https://pgdocptbr.sourceforge.io/pg80/datatype.html>

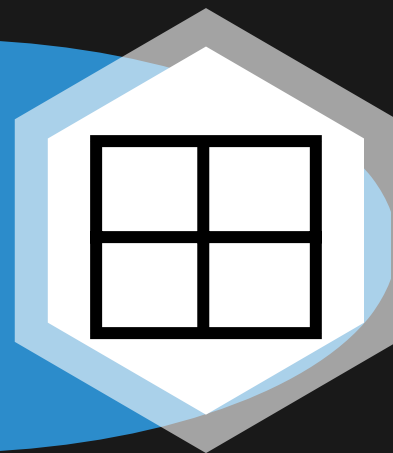
INTEGER - NÚMEROS INT.

VARCHAR - TEXTO

DATE - DATAS

PASSO 2

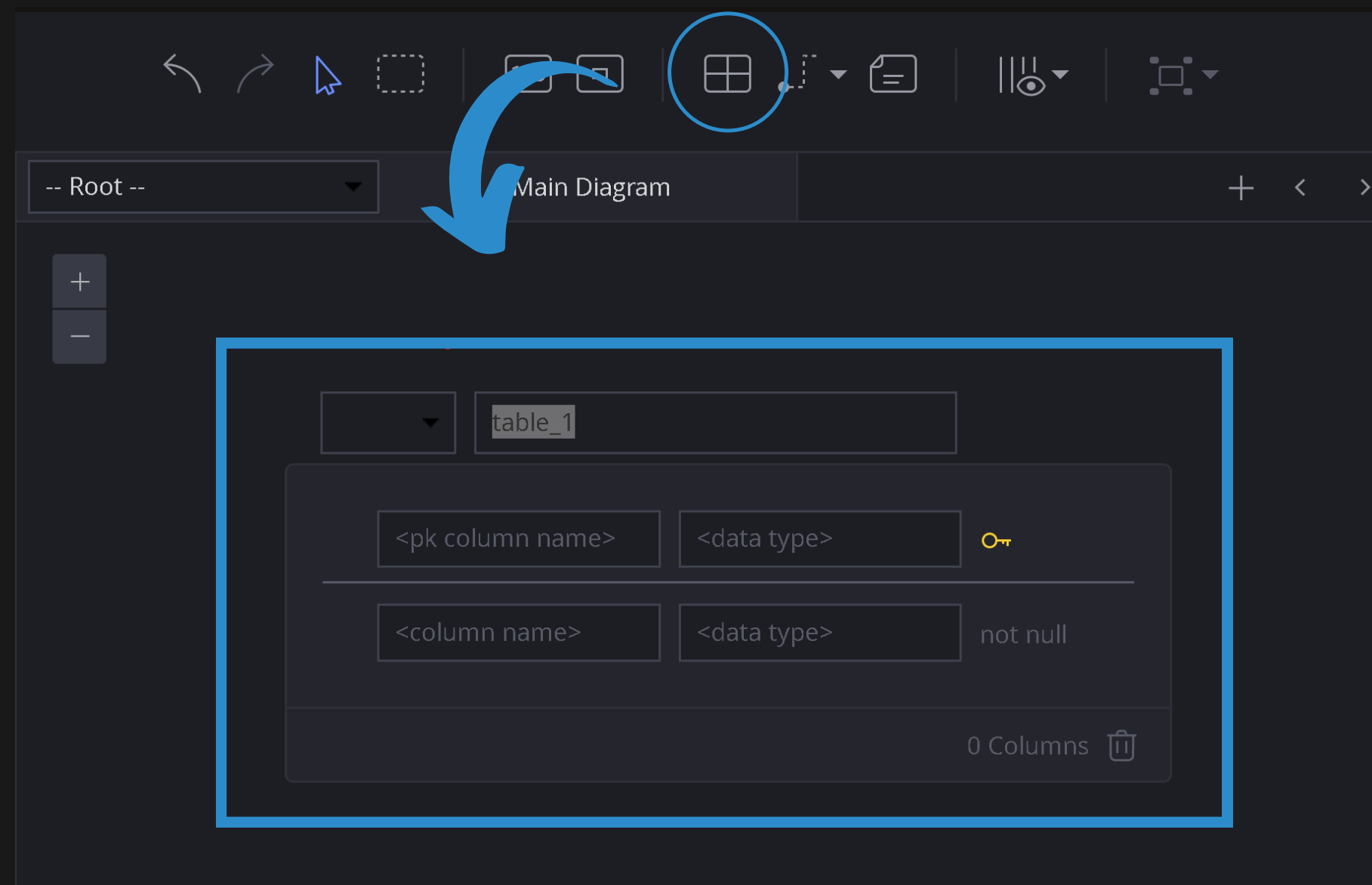
PRÁTICA



COMO INSERIR INFORMAÇÕES EM UMA TABELA

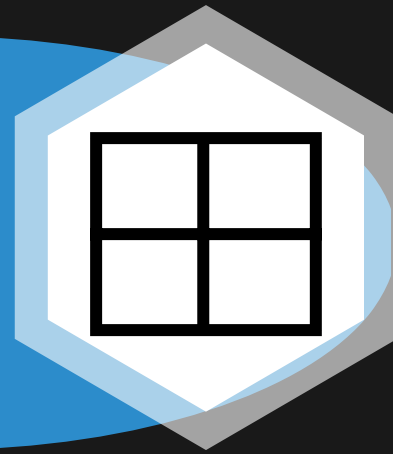
No SQLDBM

1º- CRIAR TABELA



PASSO 2

PRÁTICA



2º - ONDE INSERIR AS INFORMAÇÕES?

NOME DA ENTIDADE

CHAVES

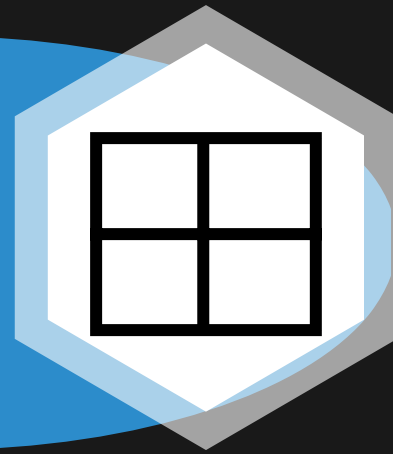
ATRIBUTOS

TIPO DE DADO

	table_1	
<pk column name>	<data type>	key
<column name>	<data type>	not null
		0 Columns

PASSO 2

PRÁTICA



EXEMPLO DE TABELA DA BIBLIOTECA

NOME DA ENTIDADE: PESSOA

CHAVES: CODIGO_PESSOA INTEGR

ATRIBUTOS:

NOME VARCHAR

TELEFONE VARCHAR

CELULAR VARCHAR

CPF VARCHAR

RG VARCHAR

▼

peessoa

⋮

codigo_pessoa

<pk column name>

<data type>

int

🔑

⋮

nome

<column name>

<data type>

varchar(50)

not null

⋮

telefone

<column name>

<data type>

varchar(50)

not null

⋮

celular

<column name>

<data type>

varchar(50)

not null

⋮

cpf

<column name>

<data type>

varchar(50)

not null

⋮

rg

<column name>

<data type>

varchar(50)

not null

<column name>

<data type>

not null

*Apesar de dados como o telefone e o rg serem numéricos, precisam ser registrados como VARCHAR por que possuem caracteres especiais (- .)

PASSO 3

CONCEITOS



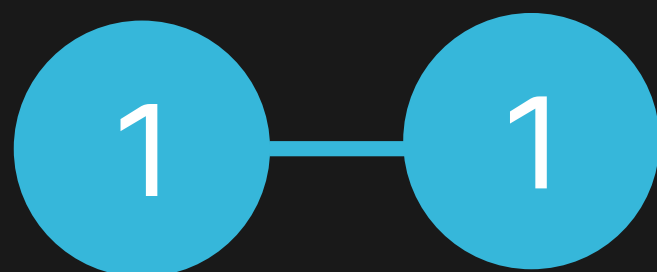
TIPOS DE RELACIONAMENTO

Agora que já temos as nossas informações jogadas nas tabelas, é possível notar que **alguns atributos ou chaves aparecem em mais de uma entidade.**

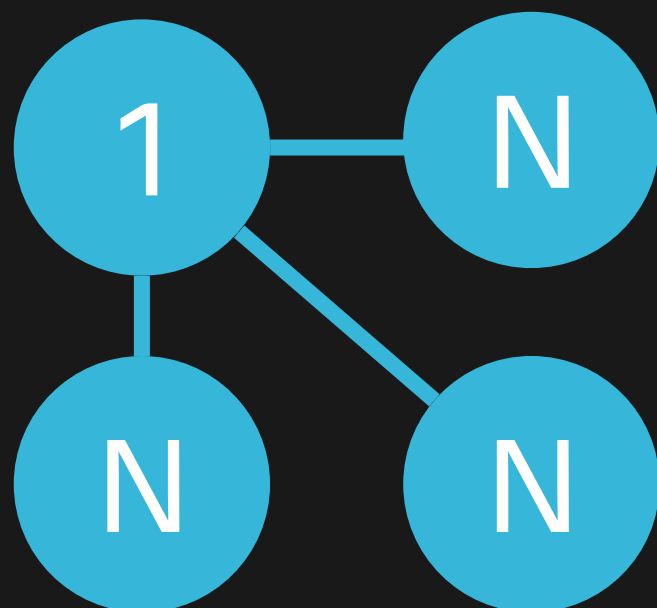
Ex: código_pessoa é uma chave presente nas entidades: pessoa, aluno e professor (visto que os 3 são pessoas);

Para conectar essas informações umas às outras precisamos estabelecer relacionamentos entre entidades diferentes.

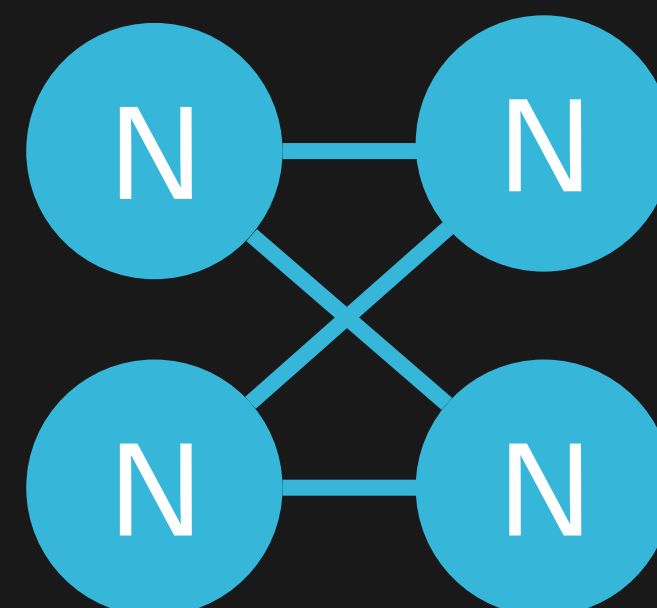
Existem três tipos principais de relacionamentos em um banco de dados: 1-1, 1-N e N-N



Nesse tipo de relacionamento, **uma única linha em uma tabela está associada a uma única linha em outra tabela.**



Nesse tipo de relacionamento, **uma única linha em uma tabela está associada a várias linhas em outra tabela.**



Nesse tipo de relacionamento, **várias linhas em uma tabela estão associadas a várias linhas em outra tabela.** Isso é geralmente implementado por meio de uma tabela intermediária que associa as duas tabelas principais.

PASSO 3

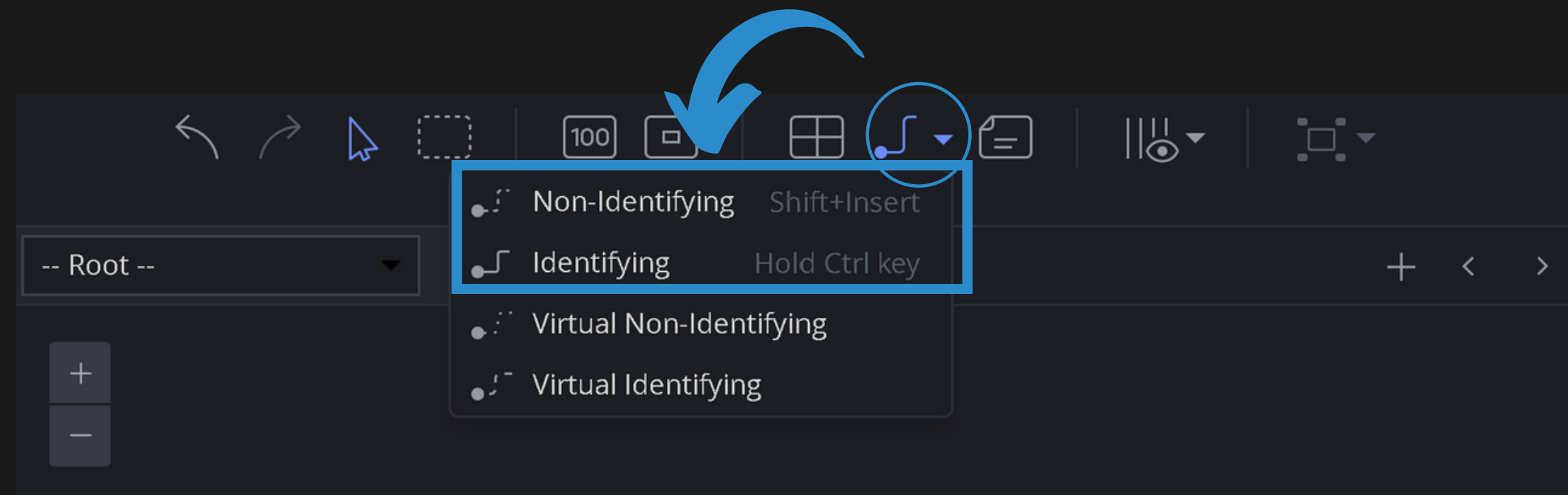
PRÁTICA



COMO ESTABELECECER UM RELACIONAMENTO ENTRE DUAS TABELAS

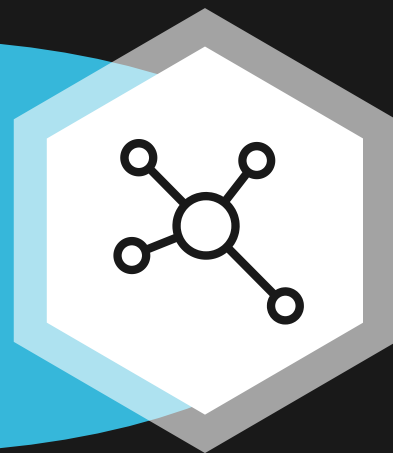
No SQLDBM

**1º SELECIONAR
"CONEXÕES" E
ESCOLHER O
TIPO DE
CONEXÃO**



PASSO 3

PRÁTICA



IDENTIFYING

Uma referência identificadora ocorre quando uma tabela referenciada (também chamada de tabela "filha") depende diretamente da tabela referenciante (também chamada de tabela "mãe"). Isso significa que a tabela filha tem uma chave estrangeira que é uma parte da chave primária da tabela mãe. Em outras palavras, a tabela filha não pode existir sem a tabela mãe.

NON-IDENTIFYING

Ocorre quando a tabela filha não depende diretamente da tabela mãe. Isso significa que a tabela filha tem uma chave estrangeira que não faz parte da chave primária da tabela mãe. Nesse caso, a tabela filha pode existir independentemente da tabela mãe.

PASSO 3

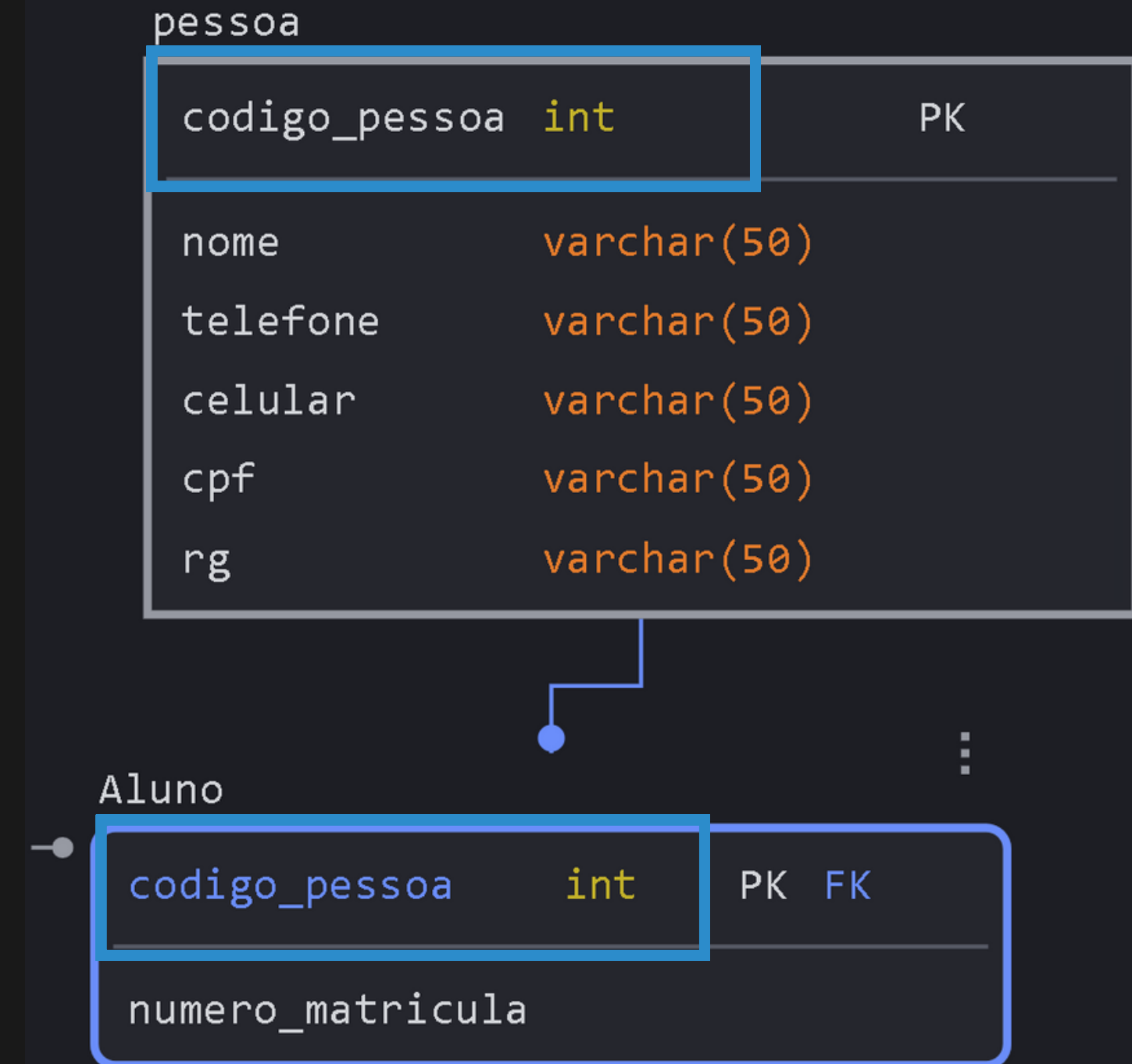
PRÁTICA



2º - SELECIONAR A CHAVE NA TABELA MÃE E A TABELA FILHA PARA ONDE VAI

PK - Primary Key (chave primária)

FK - Foreign Key (chave estrangeira)

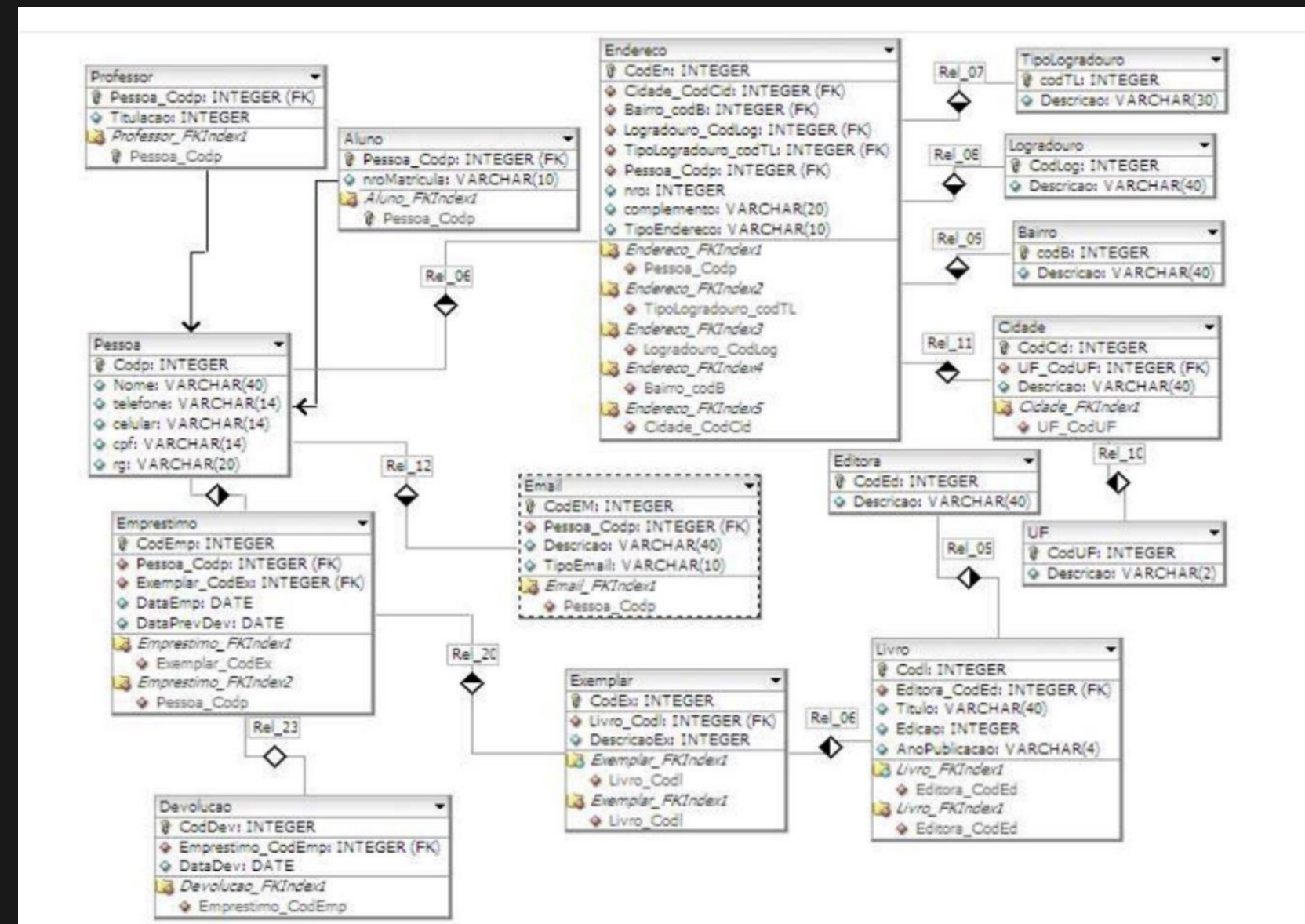


PASSO 5

PRÁTICA



COMO INTERPRETAR O MODELO DO PROFESSOR



PASSO 5

PRÁTICA



TABELAS

Emprestimo	
✓ CodEmp:	INTEGER
✱ Pessoa_Codp:	INTEGER (FK)
✱ Exemplar_CodEx:	INTEGER (FK)
✱ DataEmp:	DATE
✱ DataPrevDev:	DATE
Emprestimo_FKIndex1	
✱ Exemplar_CodEx	
Emprestimo_FKIndex2	
✱ Pessoa_Codp	

NOME DA ENTIDADE

CHAVES

ATRIBUTOS

TIPO DE DADO

TIPO DE CHAVE (FK OU PK)

INFORMAÇÕES DE CHAVES

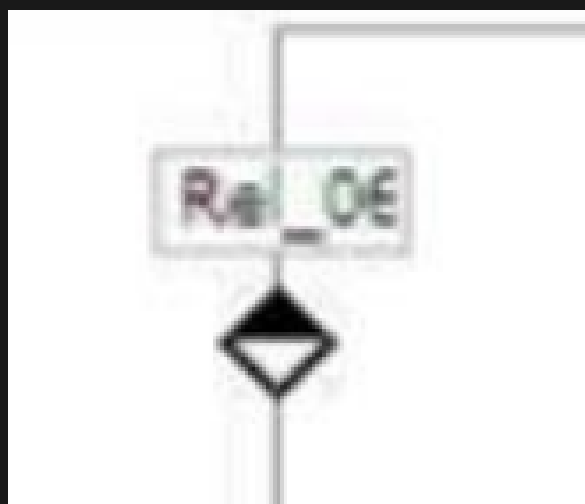
ESTRANGEIRAS

PASSO 5

PRÁTICA



RELACIONAMENTOS



IDENTIFYING

A parte pintada do losango aponta para a tabela filha (a parte vazia para a tabela mãe)

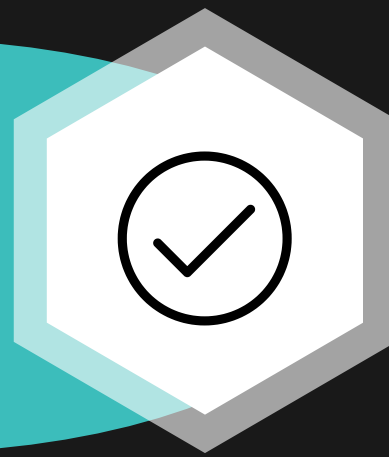


NON-IDENTIFYING

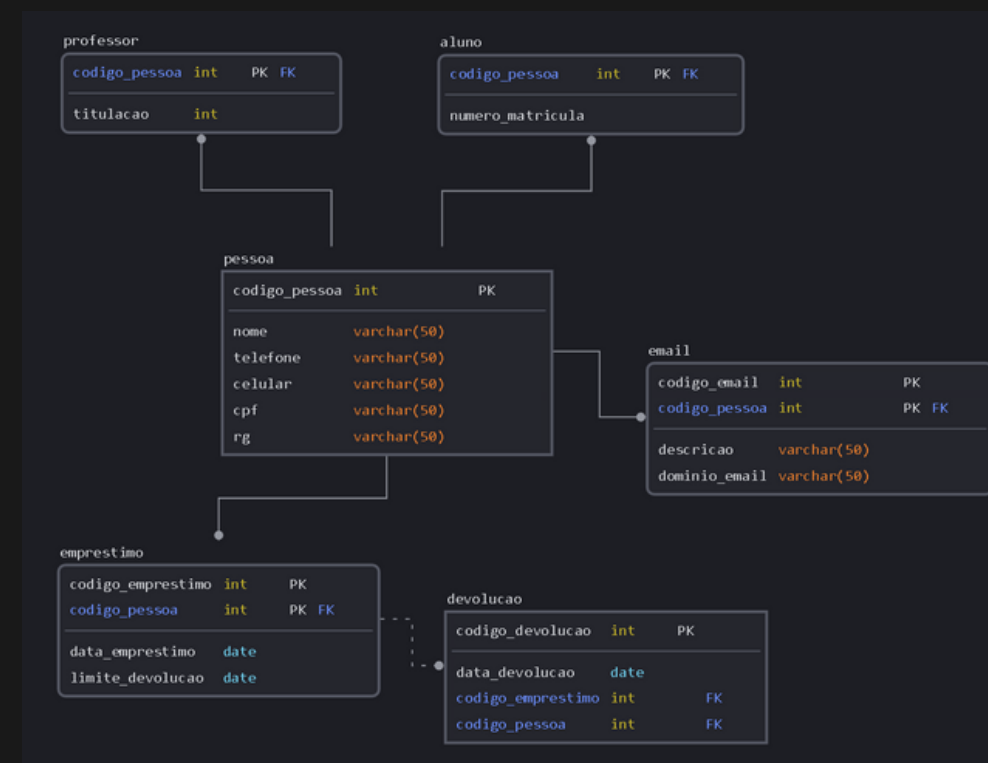
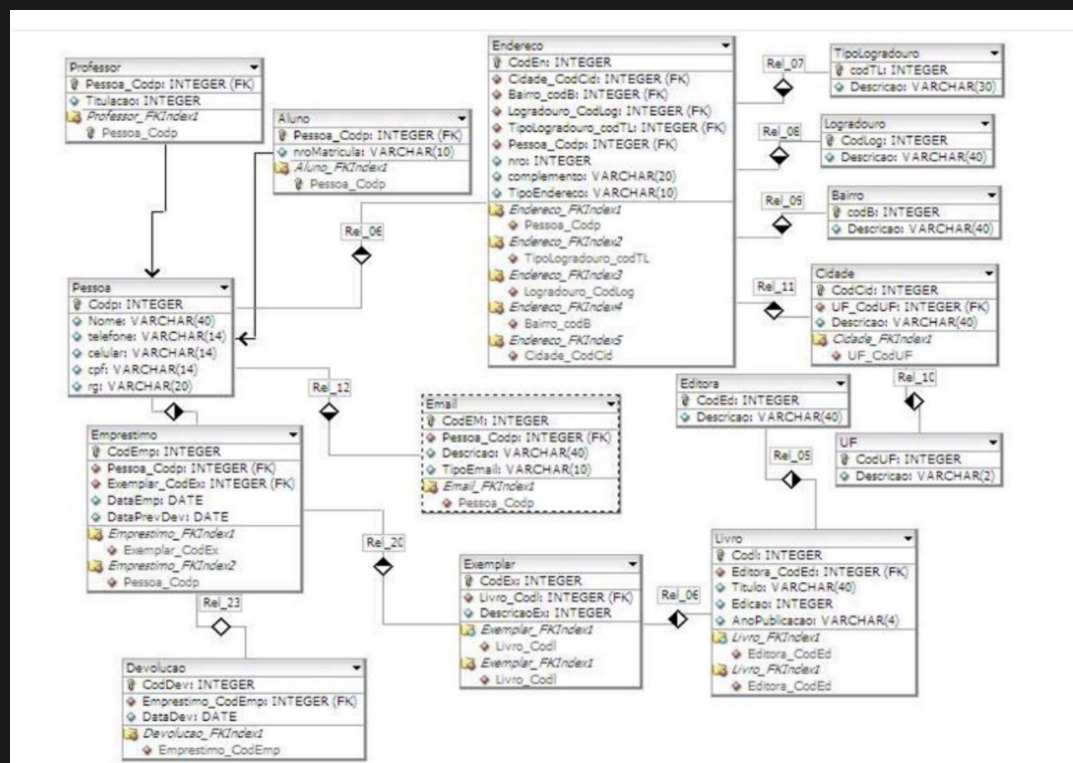
O losango não tem pintura já que a existência de uma não depende da outra

PASSO 5

PRÁTICA



Agora é só criar as mesmas tabelas da referência do professor no sqlbdm e conectá-las conforme o necessário (e se te fizer sentido, adicionar outros elementos)



Depois que terminar é só tirar um print da tela e selecionar a opção "Forward Engineer" para exportar a DDL que o professor solicitou

BOA SORTE RAPEIZE!