

# Transformações ER para MR e dicionário de dados

Na aula anterior, você aprendeu como mapear alguns conceitos do Modelo Entidade Relacionamento (ER) para o Modelo Relacional (MR). Nesta aula, você vai estudar como mapear os relacionamentos no modelo ER para o MR. Além disso, você vai estudar também os dicionários de dados e verá como eles podem ser úteis para documentar seu modelo relacional.

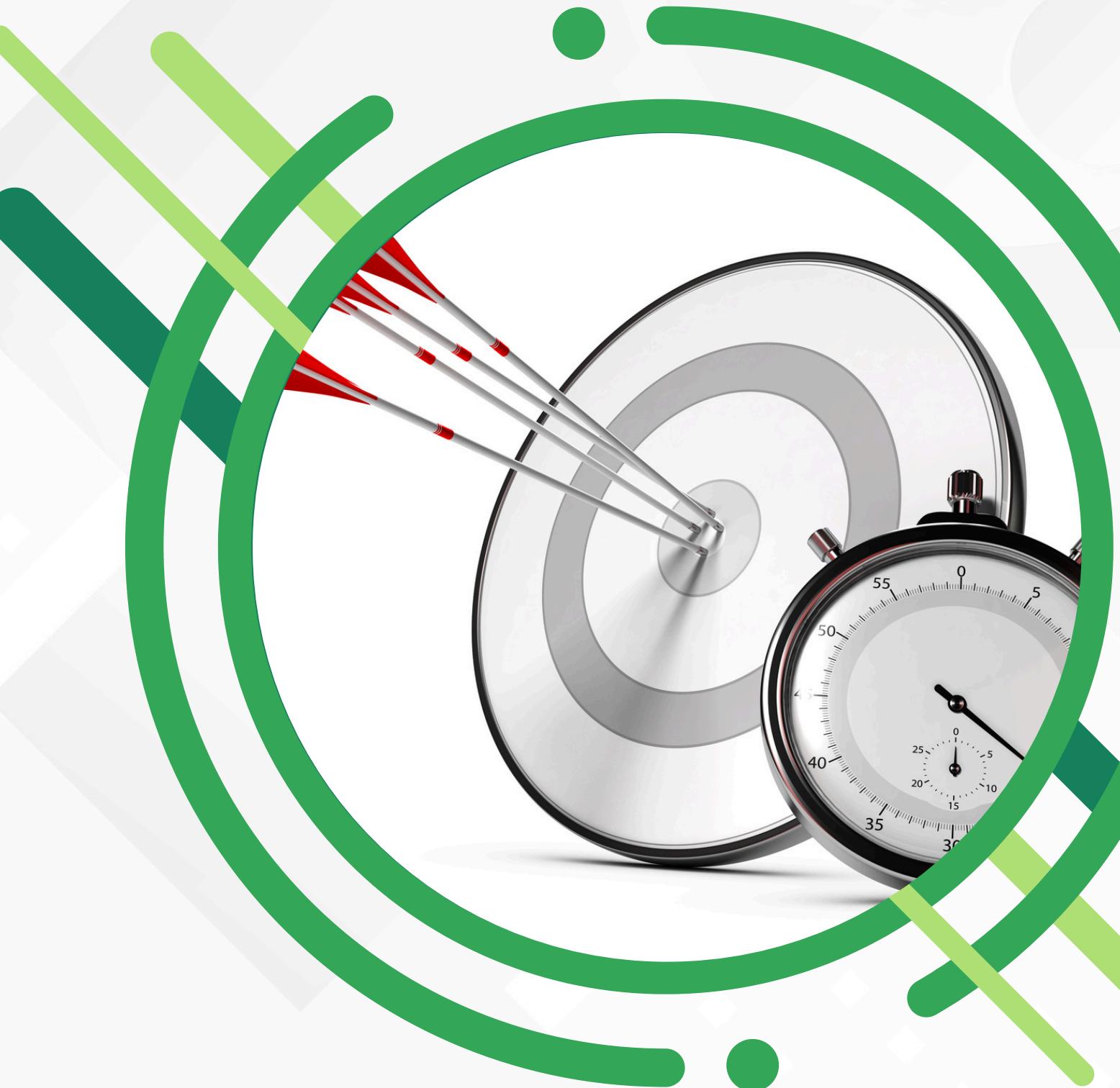
professor

**Paulo Perris**



# Objetivos

- Transformar relacionamentos no modelo ER para o Modelo Relacional.
- Utilizar dicionário de dados para melhorar a documentação do Modelo Relacional.



# Transformações ER-MR



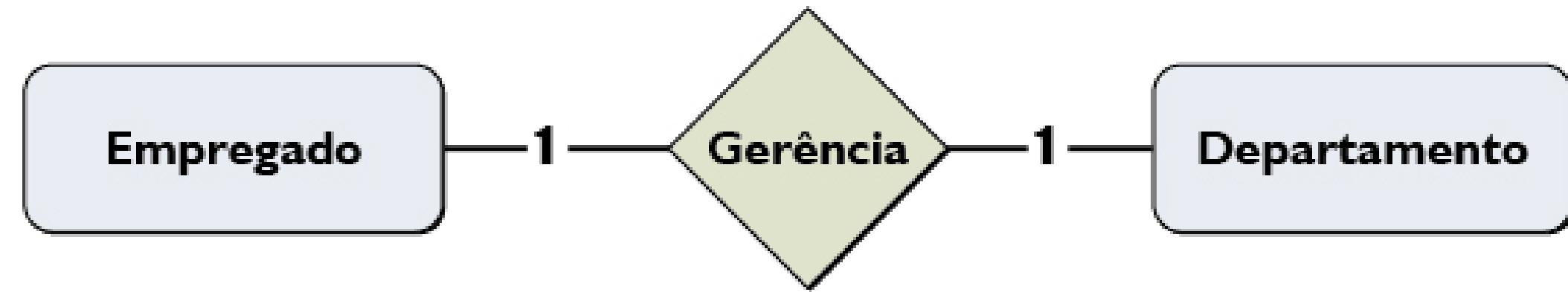
Como você estudou na aula anterior, o processo de transformação do Modelo Entidade Relacionamento (MER) para o Modelo Relacional (MR) envolve alguns passos. Você aprendeu como mapear entidades e os vários tipos de atributos existentes. Na última aula, paramos no quinto passo, que faz o mapeamento de um atributo multivalorado do modelo ER para o Modelo Relacional.

Nesta aula, daremos continuidade aos passos da aula anterior, retomando pelo sexto passo, mas agora fazendo as transformações dos tipos de relacionamentos existentes. Vamos lá?

# Sexto passo: mapear relacionamento um-para-um

Depois de mapear os atributos do modelo ER, agora você vai mapear os relacionamentos. O primeiro tipo de relacionamento é o um-para-um.

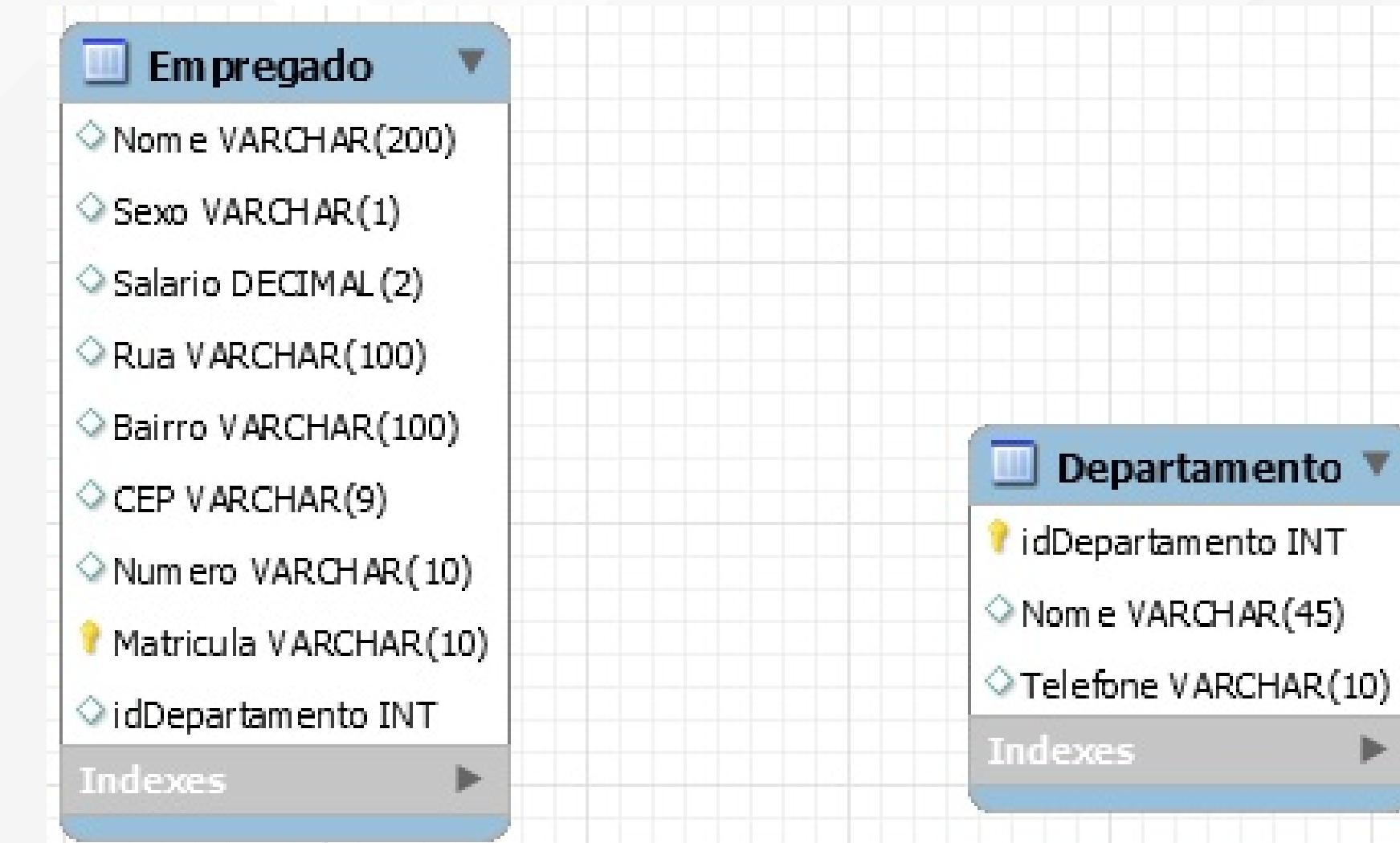
A Figura 1 demonstra um exemplo de relacionamento um-para-um no modelo ER.



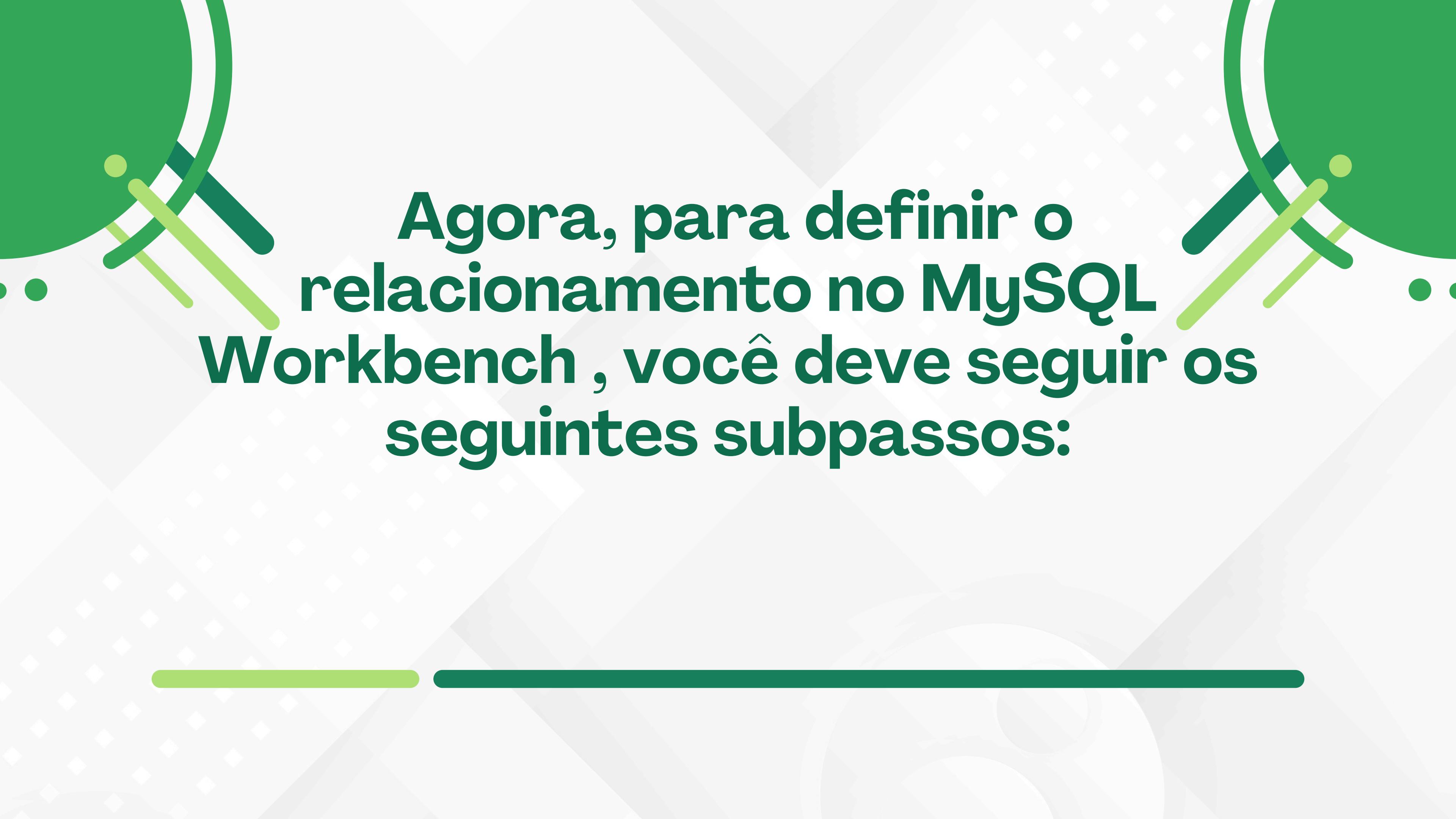
# Sexto passo: mapear relacionamento um-para-um

Para mapear este relacionamento para o Modelo Relacional, devemos inserir em uma tabela uma chave estrangeira que referencia a chave primária da outra entidade.

Para o caso do exemplo, devemos definir na tabela Empregado uma chave estrangeira que referencia a chave primária da tabela Departamento.



Por exemplo, a Figura 2 mostra as duas tabelas sem o relacionamento um-para-um.



**Agora, para definir o  
relacionamento no MySQL  
Workbench , você deve seguir os  
seguintes subpassos:**

# Sexto passo: mapear relacionamento um-para-um

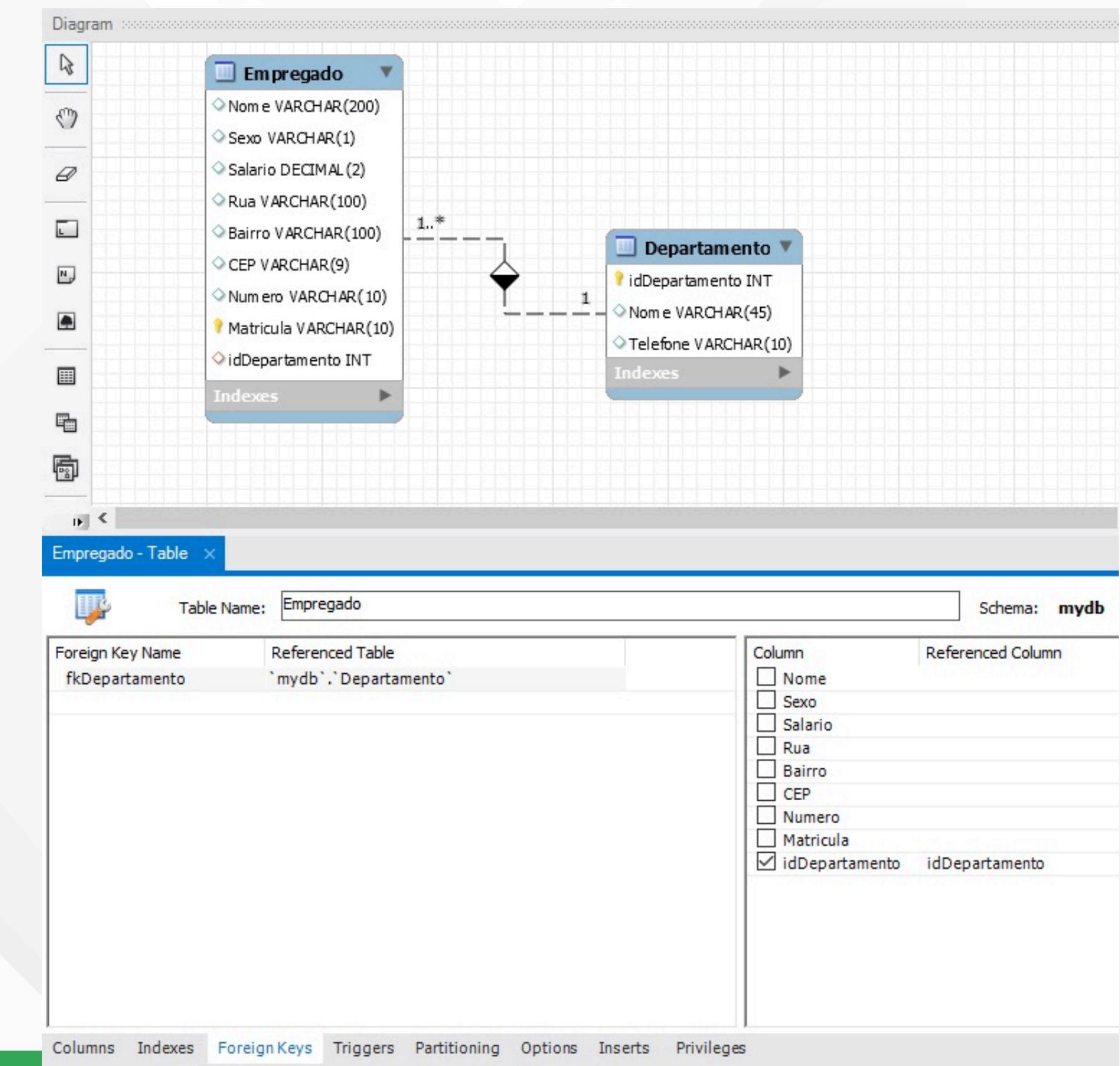
Inserir um atributo na tabela Empregado com o mesmo nome e tipo do atributo chave primária da tabela Departamento, neste caso, idDepartamento. Ao final deste subpasso, você deve obter algo parecido com a Figura 3.

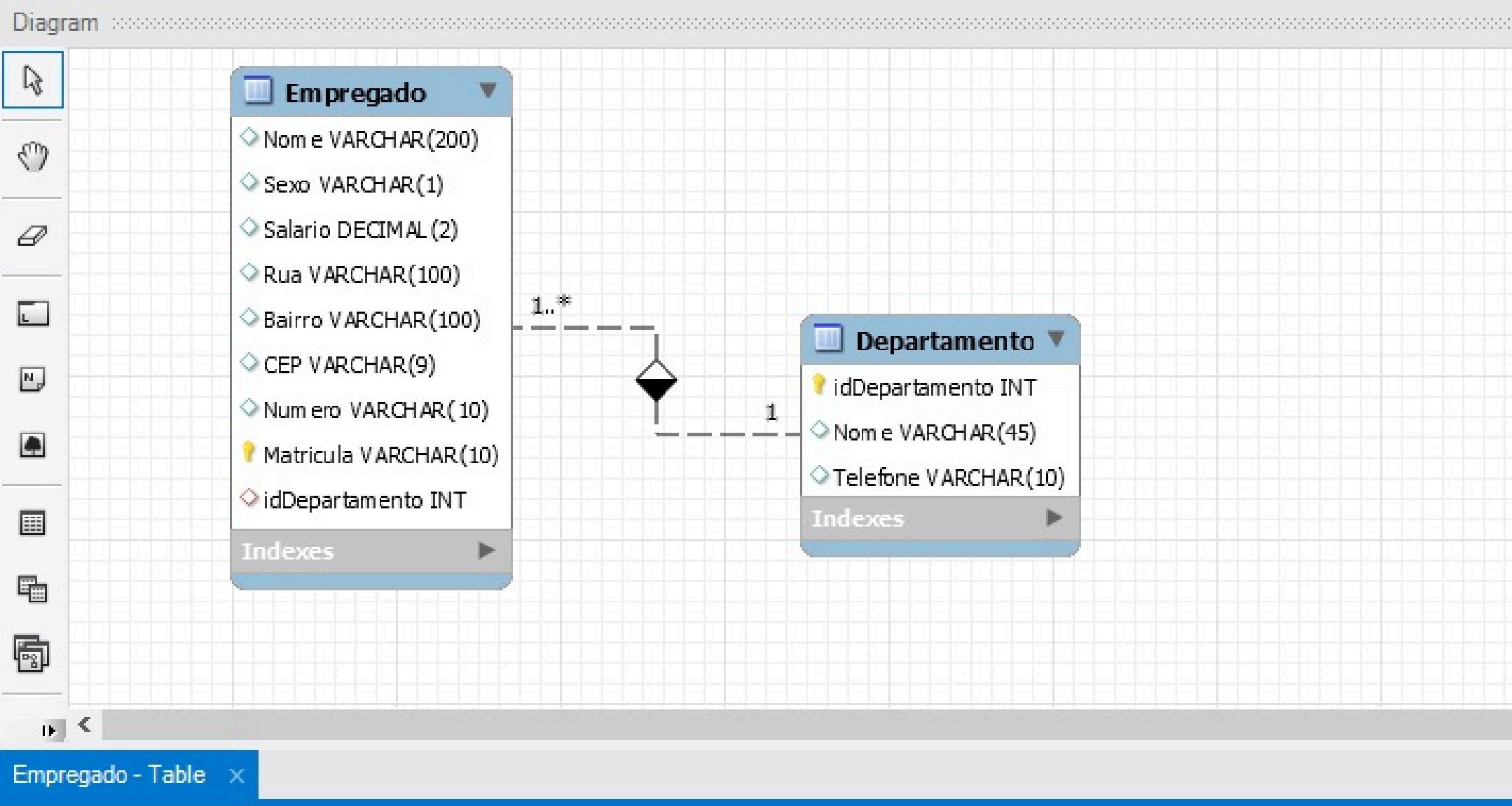
Empregado	
◊	Nome VARCHAR(200)
◊	Sexo VARCHAR(1)
◊	Salario DECIMAL(2)
◊	Rua VARCHAR(100)
◊	Bairro VARCHAR(100)
◊	CEP VARCHAR(9)
◊	Numero VARCHAR(10)
!	Matricula VARCHAR(10)
◊	idDepartamento INT
Indexes	

# Sexto passo: mapear relacionamento um-para-um

Definir uma chave estrangeira com o atributo idDepartamento para referenciar a chave primária da tabela Departamento.

Ao final deste subpasso, você deve obter algo parecido com a Figura 4.





Empregado - Table

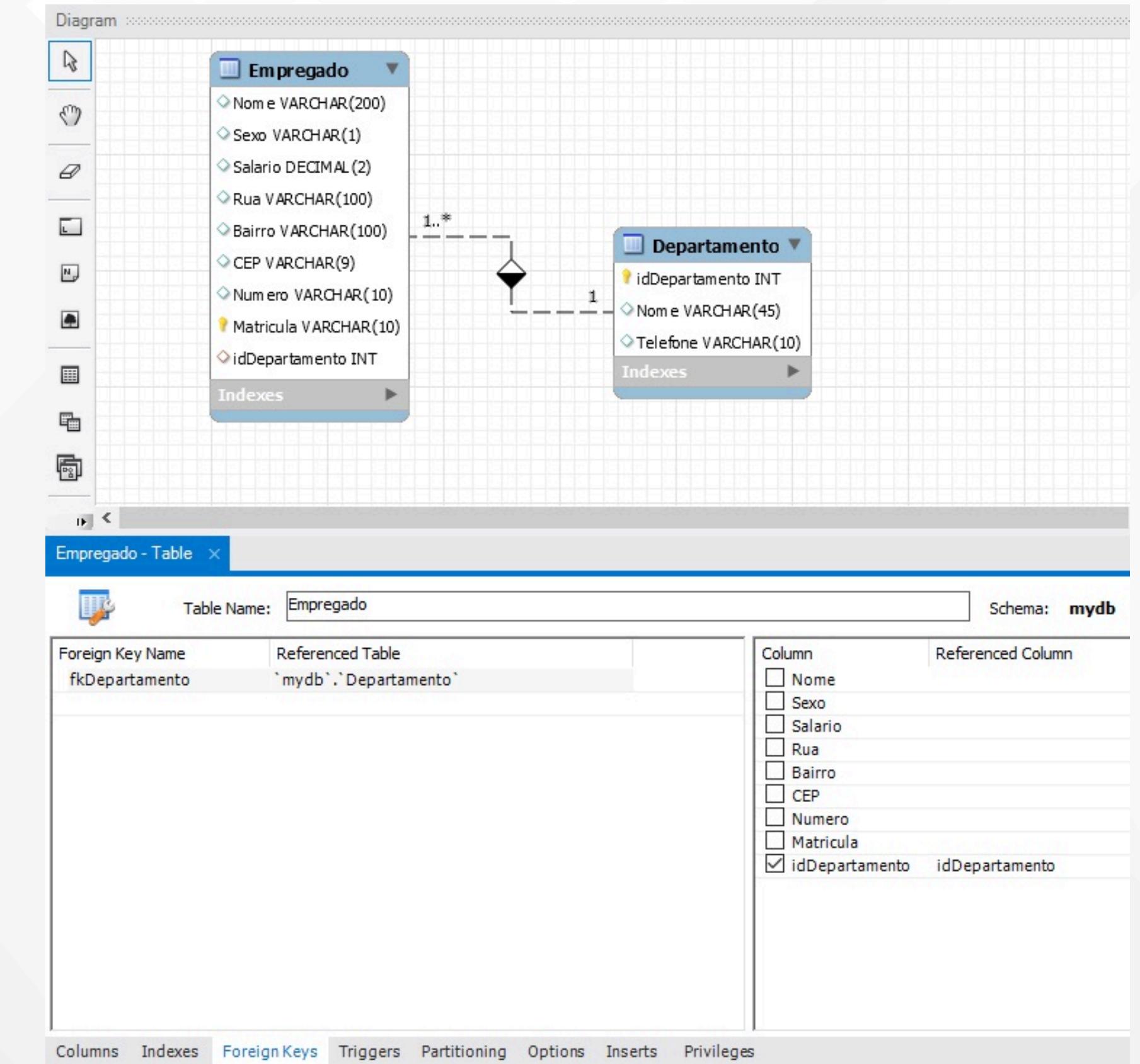
Table Name: Empregado		Schema: mydb
Foreign Key Name	Referenced Table	Column
fkDepartamento	'mydb'.'Departamento'	<input type="checkbox"/> Nome
		<input type="checkbox"/> Sexo
		<input type="checkbox"/> Salario
		<input type="checkbox"/> Rua
		<input type="checkbox"/> Bairro
		<input type="checkbox"/> CEP
		<input type="checkbox"/> Numero
		<input type="checkbox"/> Matricula
		<input checked="" type="checkbox"/> idDepartamento    idDepartamento

Columns   Indexes   **Foreign Keys**   Triggers   Partitioning   Options   Inserts   Privileges

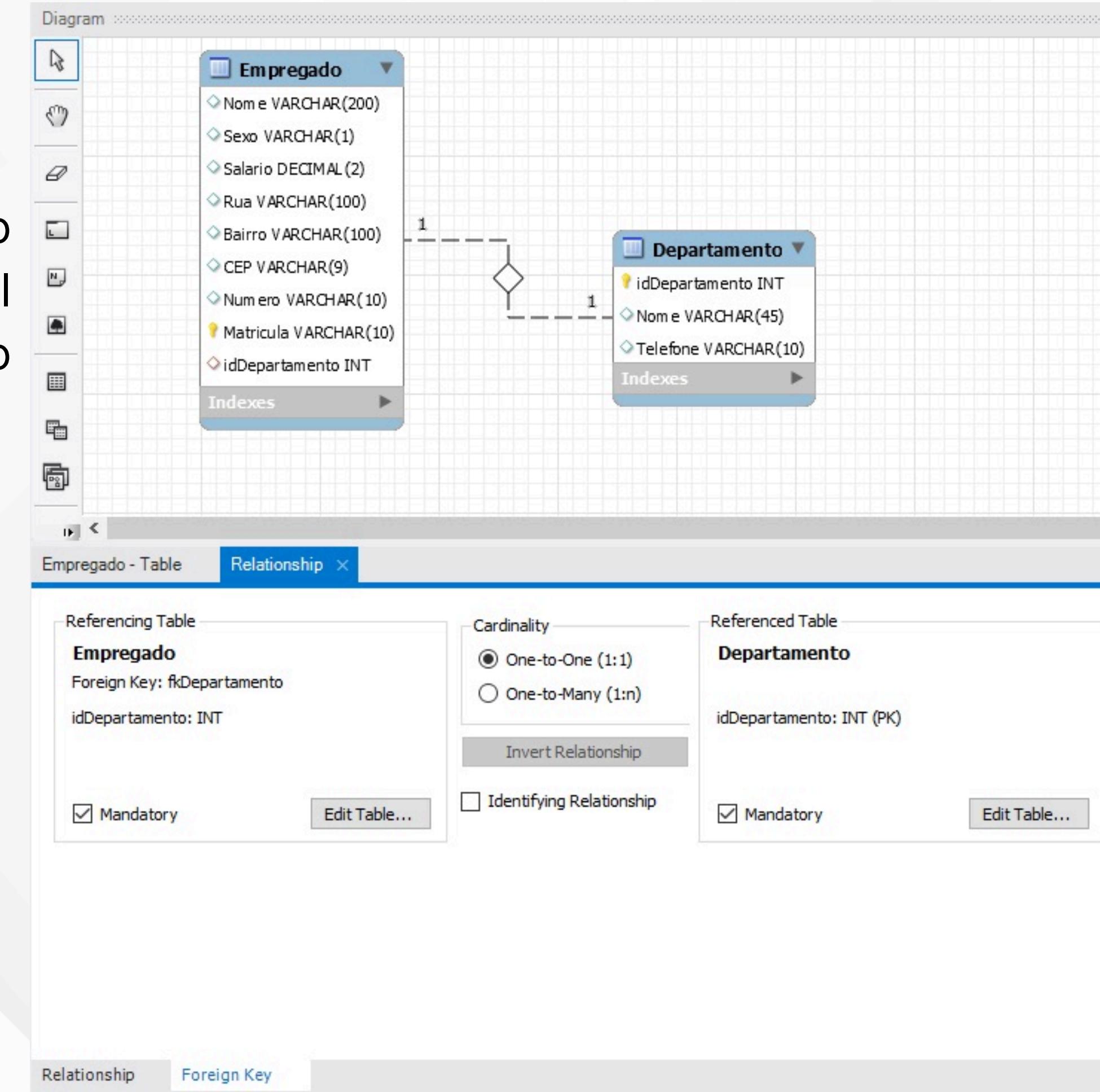
Note que na Figura 4 aparece a cardinalidade de 1..\* e 1, ou seja, o MySQL Workbench cria por padrão um relacionamento um-para-muitos.

Portanto, devemos ajustar o relacionamento criado, através de um clique com o botão direito do mouse em cima da linha que interliga as duas tabelas presentes na Figura 4.

Escolha a opção Edit Relationship.

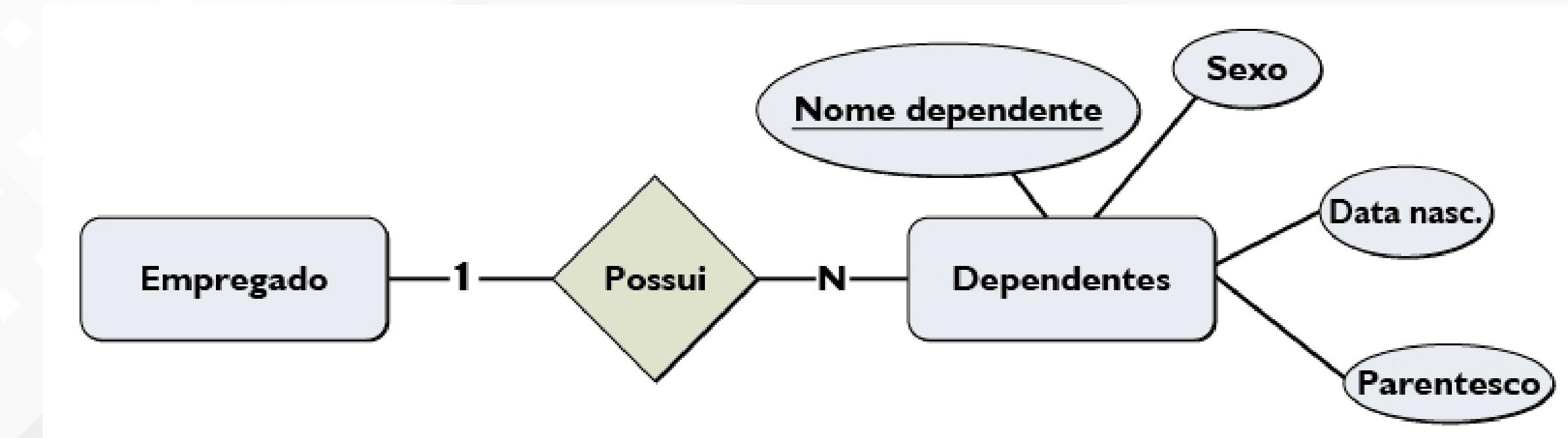


Na aba Foreign key, marque a opção Cardinality para One-to-one (1:1). Ao final deste subpasso, você deve obter algo parecido com a Figura 5.



# Sétimo passo: mapear relacionamento um-para-muitos

Você se lembra da nossa de MER? Nela, vimos que um relacionamento um-para-muitos é usado quando uma entidade A pode se relacionar com uma ou mais entidades B. A Figura 6 mostra um relacionamento um-para-muitos no modelo ER.



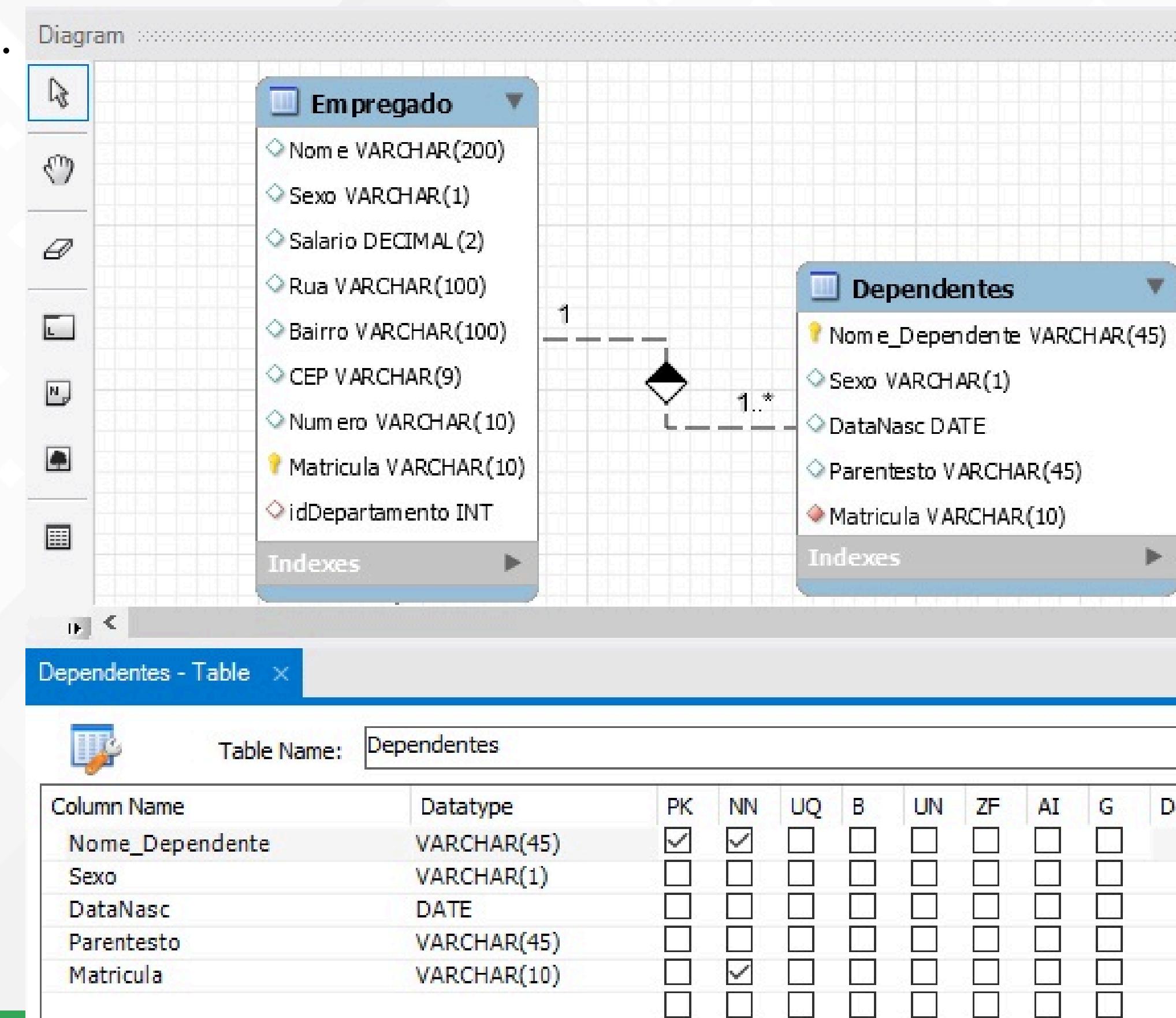
Agora veremos como mapear tal relacionamento para o Modelo Relacional. Para definir o relacionamento no MySQL Workbench , você deve primeiro mapear as entidades Empregado e Dependentes para o Modelo Relacional, como já vimos anteriormente.

Depois você deve realizar os seguintes subpassos:

1. Inserir um atributo na tabela **Dependentes** com o mesmo nome e tipo do atributo chave primária da tabela Empregados, neste caso, o atributo Matricula. Uma dúvida que você pode ter é “como saber onde criar (qual tabela) o novo atributo?”. A resposta para esta dúvida é a seguinte: sempre que você tiver uma relação um-para-muitos ou 1:N, a entidade que estiver do lado N deverá receber o novo atributo com a chave estrangeira. No caso do exemplo, a entidade que está do lado do N é a entidade Dependentes. Ao final deste subpasso, você deve obter algo parecido com a Figura 7.

Dependente	
Nome_Dependente	VARCHAR(45)
Sexo	VARCHAR(1)
DataNasc	DATE
Parentesco	VARCHAR(45)
Matricula	VARCHAR(10)

Definir uma chave estrangeira com o atributo Matricula para referenciar a chave primária da tabela Empregado. Ao final deste subpasso, você deve obter algo parecido com a Figura 8.



Note que a tabela Dependente possui como chave primária o atributo Nome\_Dependente.

Esse atributo foi criado durante o mapeamento da entidade Dependente no Modelo ER para o Modelo Relacional.

Como você sabe, um atributo chave não pode ter duas tuplas com o mesmo valor.

Neste caso, se dois empregados tiverem dois dependentes com o mesmo nome, o SGDB não irá aceitar o cadastro de um deles.

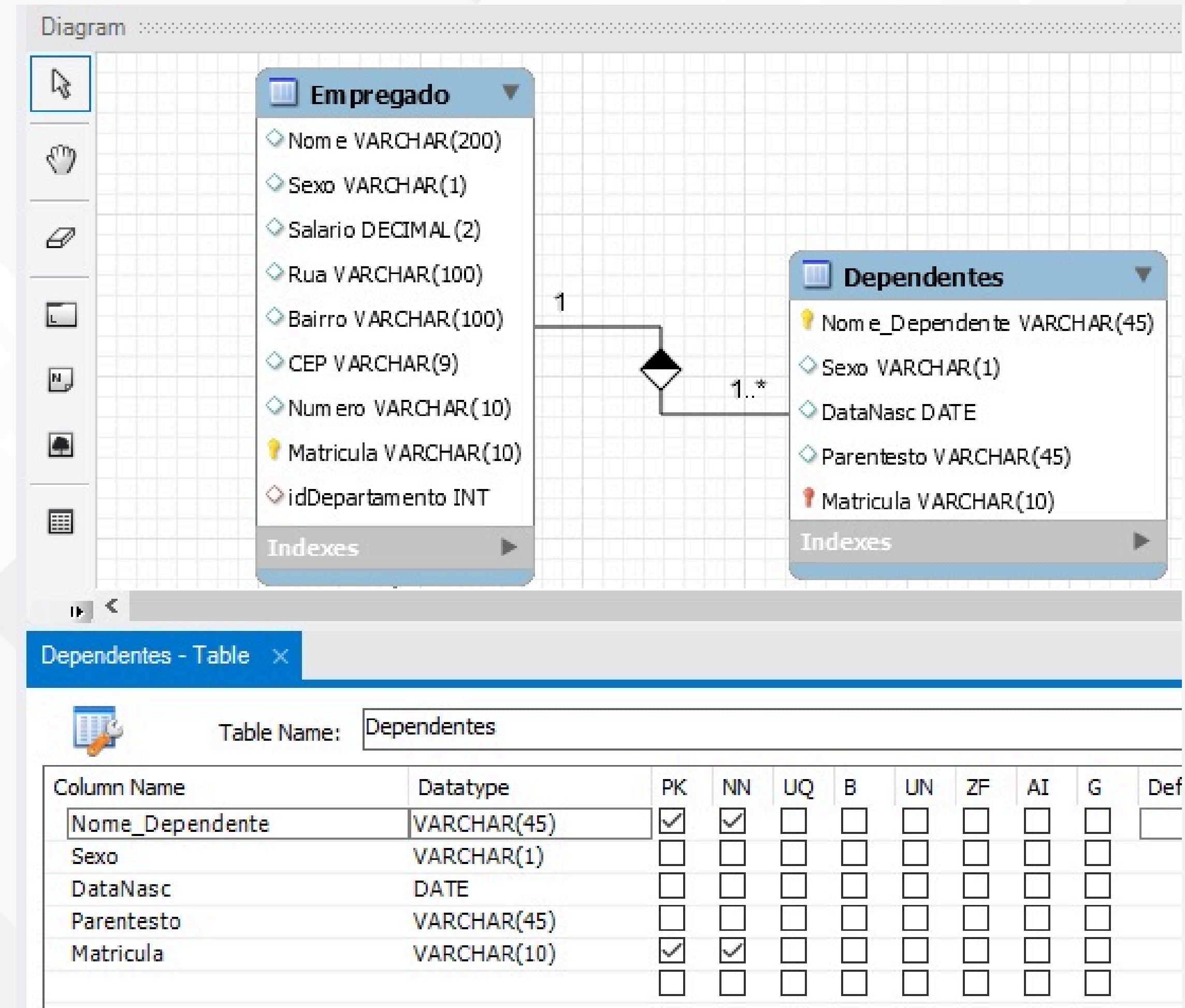
Para evitar esse problema, você deve inserir o campo adicionado pelo subpasso 1 como chave primária da tabela.

No caso, o campo adicionado foi o campo Matricula.

Assim, a tabela Dependente terá como chave primária os campos Nome\_Dependente e Matricula.

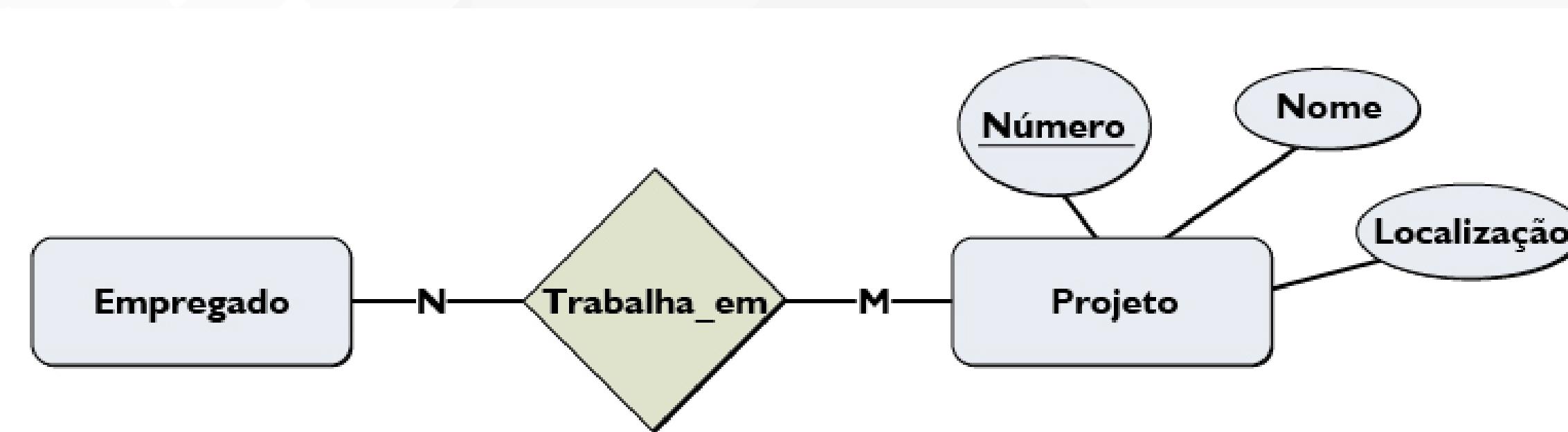
Ao final deste subpasso, você deve obter algo parecido com a Figura 9.

Note que agora dois dependentes podem ter o mesmo nome, desde que eles estejam vinculados a empregados diferentes.



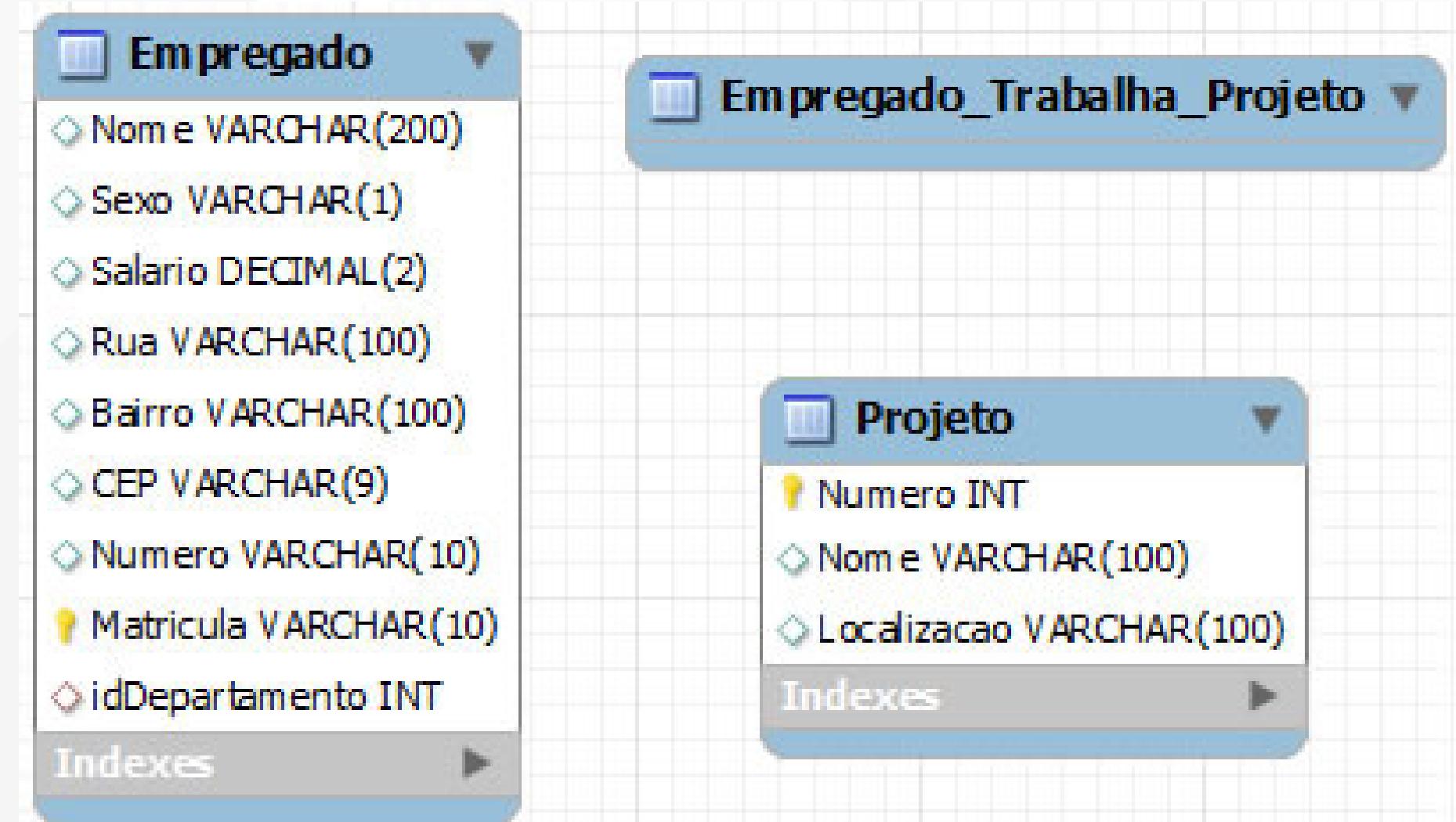
# Oitavo passo: mapear relacionamento muitos-para-muitos

Neste passo, você deve primeiro lembrar que o relacionamento muitos-para-muitos é usado quando várias entidades A se relacionam com várias entidades B. No exemplo da Figura 10 , temos o relacionamento muitos-para-muitos entre Empregado e Projeto. Nela, a entidade Empregado trabalha em vários (M) Projetos. Por outro lado, cada projeto possui (N) empregados.

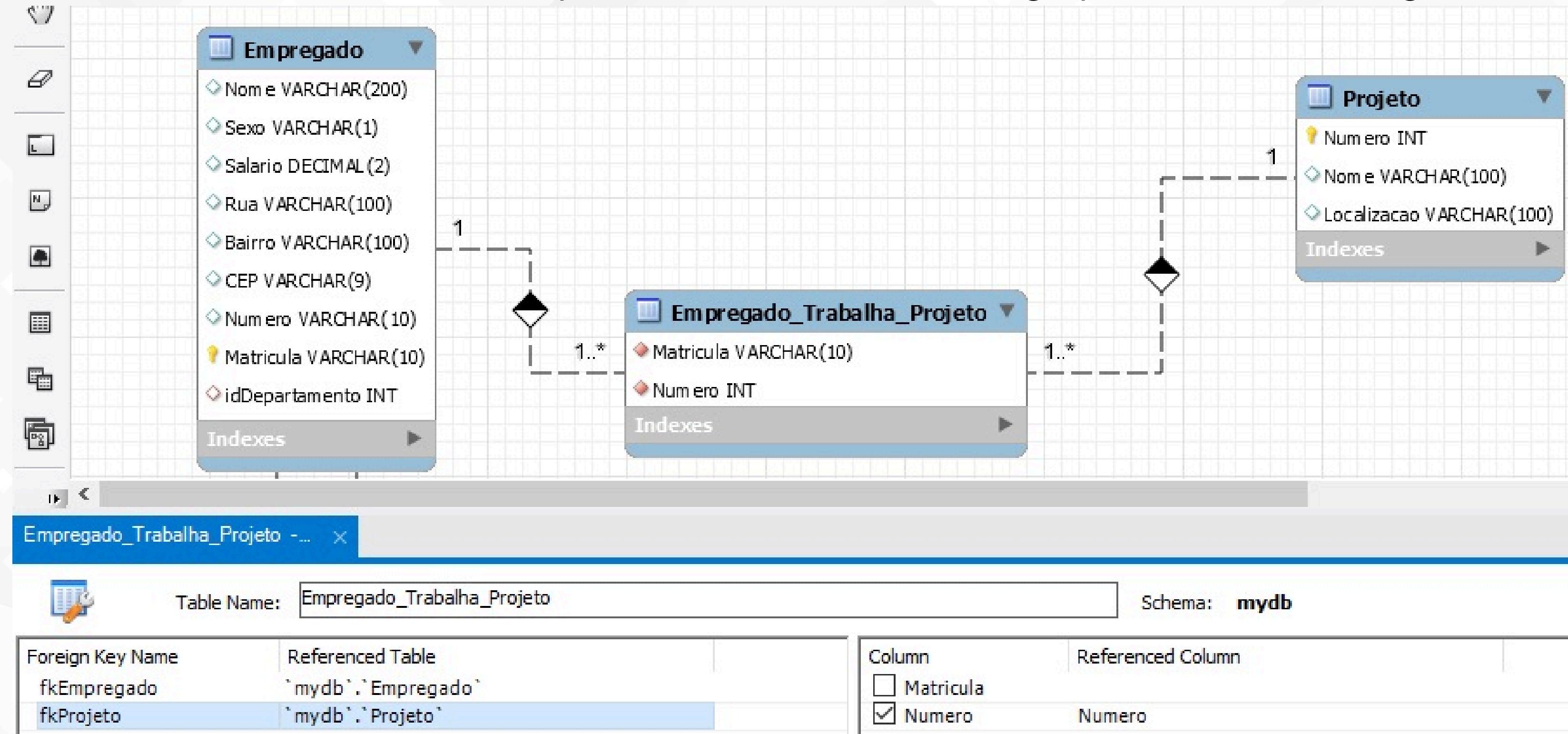


Para definir o relacionamento muitos-para-muitos no Modelo Relacional, usando o MySQL Workbench , você deve primeiro mapear as entidades Empregado e Projeto para o Modelo Relacional, como já mostramos anteriormente.

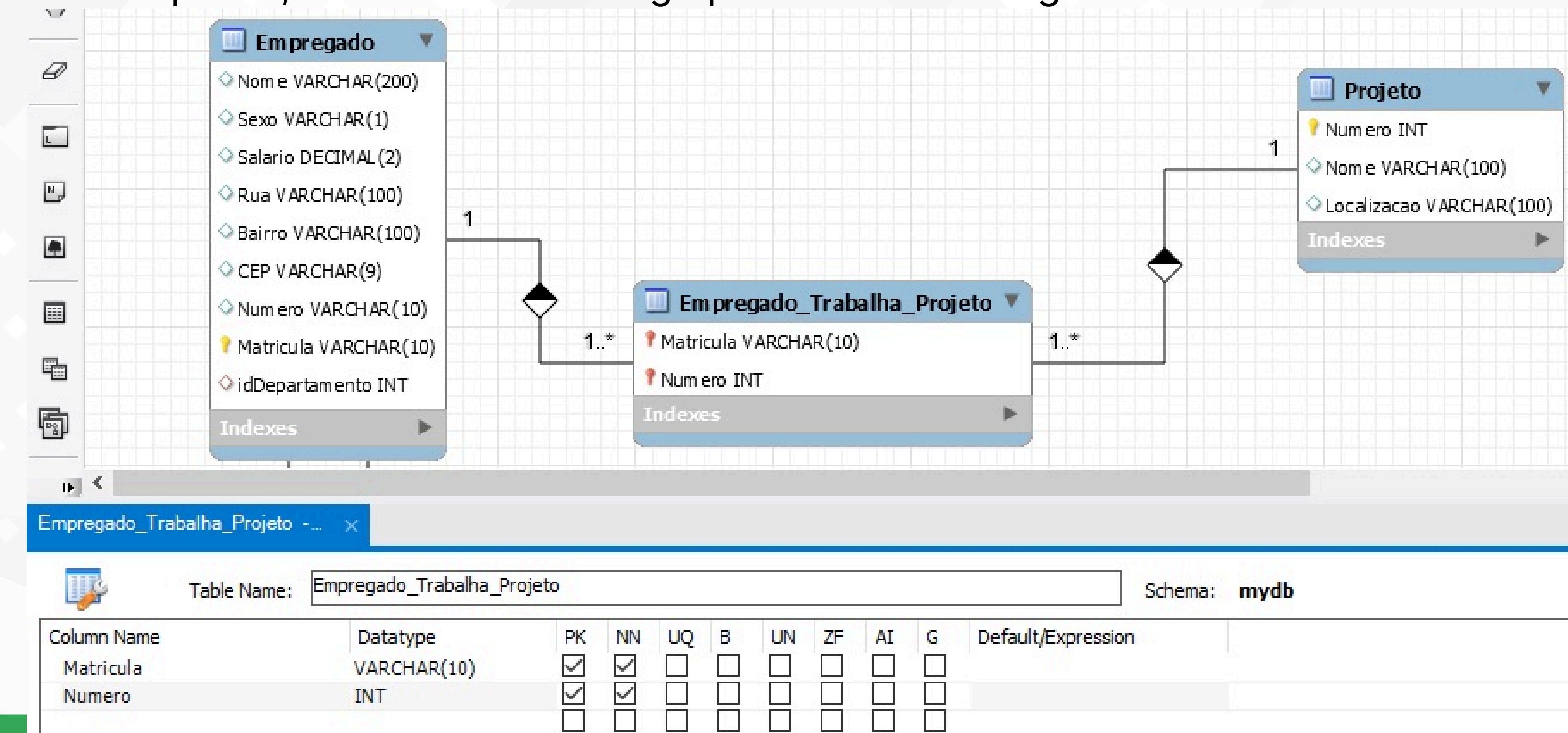
Criar uma nova tabela para representar o relacionamento muitos-para-muitos. No nosso exemplo, você poderia criar uma tabela com o nome Empregado\_Trabalha\_Projeto para representar o relacionamento no modelo ER. Ao final deste subpasso, você deve obter algo parecido com a Figura 11.



Inserir, como chave estrangeira na tabela recém-criada, as chaves primárias das entidades participantes. No caso do exemplo, você deve inserir como chave estrangeira os atributos Matrícula e Número, que são as chaves primárias das tabelas Empregado e Projeto, respectivamente. Ao final deste subpasso, você deve obter algo parecido com a Figura 12.



O último passo consiste em definir a chave primária da tabela criada para representar o relacionamento muitos-para-muitos. A chave primária da tabela criada será a composição das chaves primárias das tabelas participantes da relação. No caso do exemplo, você deve definir como chave primária da tabela Empregado\_Trabalha\_Projeto os campos Matricula e Numero. Ao final deste subpasso, você deve obter algo parecido com a Figura 13.



# Dicionário de dados

Nas duas Aulas anteriores, você aprendeu como mapear os conceitos do modelo ER para o Modelo Relacional. Em geral, um problema que pode acontecer quando se usa o Modelo Relacional é a impossibilidade de documentar o significado dos atributos, chaves primárias e estrangeiras de uma tabela. Para resolver esse problema, você deve sempre utilizar um **dicionário de dados**.



O dicionário de dados descreve a terminologia utilizada para o desenvolvimento do modelo de dados do sistema. Ele apresenta uma descrição textual da estrutura lógica e física do banco de dados.

Em um dicionário de dados, você pode simplesmente colocar a descrição de campos e tabelas por extenso, como também colocar outras características dos campos, como tipo de dado, restrições, possíveis entradas, se ele é requerido ou não etc. Assim, o dicionário de dados é principalmente utilizado para documentar os modelos criados. Na sequência, você verá alguns exemplos de como dicionários de dados podem ser utilizados para documentar tabelas, relações e atributos.



# Representando tabelas

Para representar uma tabela em um dicionário de dados, vamos pegar como exemplo a entidade Projeto. Você pode usar o modelo a baixo.

Nome: Projeto

Descrição: Qualquer projeto realizado por empregados da empresa

Atributo chave: Número

Outros atributos: Nome e localização

# Representando relações

Todas as relações precisam ser documentadas em um modelo de dados. Cada relação recebe um nome único que será utilizado para identificar as relações. Cada relação deve ser descrita em termos de um nome, participantes, cardinalidade e uma descrição que poderá ser utilizada para facilitar o entendimento da relação no futuro.

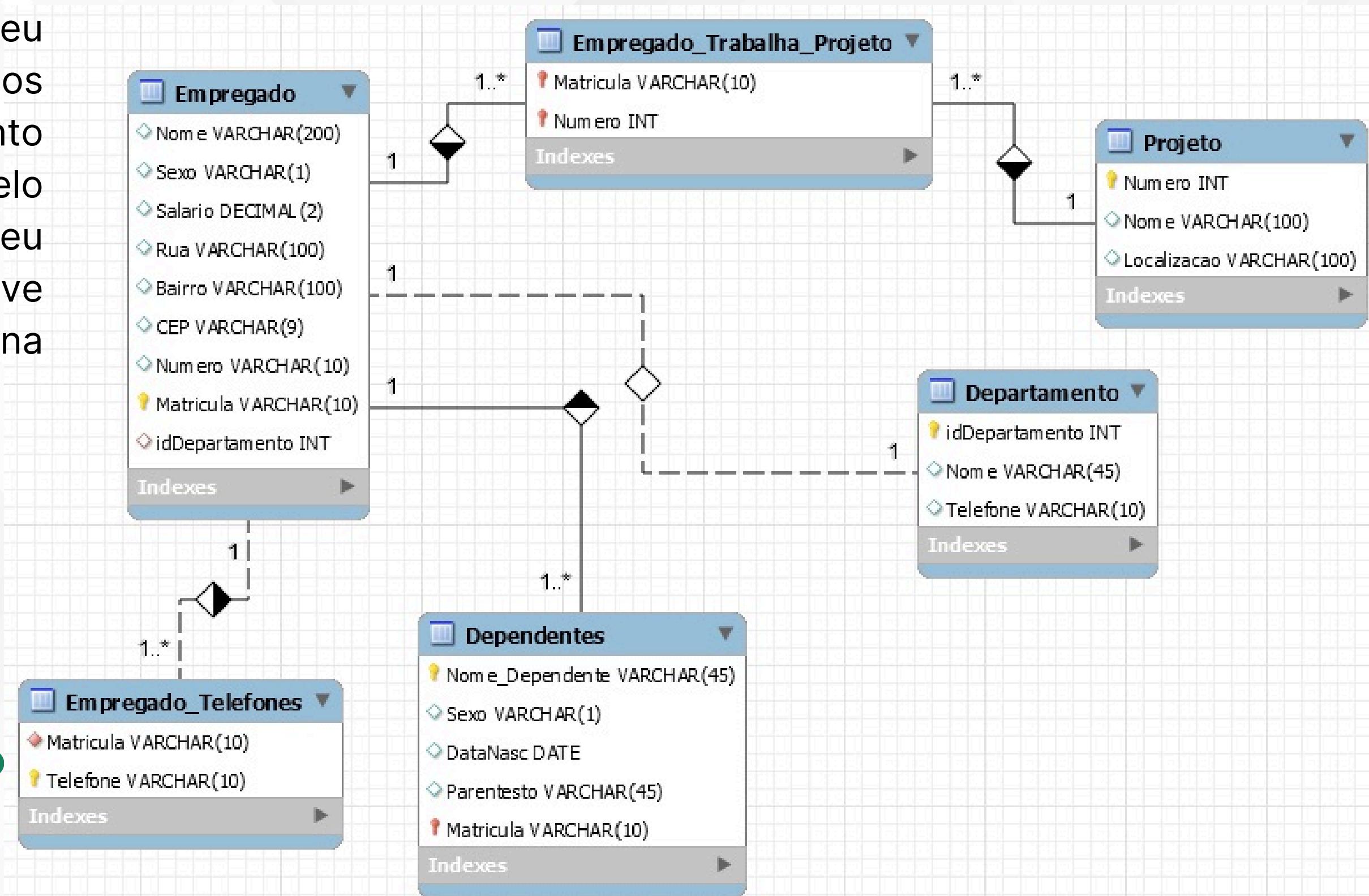
# Representando atributos

Os atributos são elementos importantes em um Modelo Relacional. Assim, eles devem ser documentados em detalhes. Para cada atributo você deve indicar o nome, uma descrição, o tipo e as possíveis restrições. Novamente, o dicionário de dados é útil para evitar que no decorrer de um projeto de desenvolvimento de software você se pergunte qual a utilidade de certo atributo. Se você tiver escrito um bom dicionário de dados, todos os seus atributos estarão documentados e, portanto, você pode consultar o dicionário para esclarecer tais dúvidas. Ainda com a entidade Projeto, o modelo ao lado representa o dicionário de dados do atributo Localização.

Nome: Localização  
Tabela: Projeto  
Descrição: Armazena onde o projeto está sendo executado  
Tipo: Caractere  
Restrição: Nenhuma

# Conclusão

Nesta quinta aula, você aprendeu como mapear os conceitos aprendidos no Modelo Entidade Relacionamento para os conceitos do Modelo Relacional. Ao final desta aula, seu diagrama no MySQL Workbench deve estar parecido com o mostrado na Figura 14.



# Resumo

Nesta aula, você aprendeu como mapear relacionamentos no modelo ER para o Modelo Relacional.

Você verificou que para cada tipo de relacionamento existe uma sequência de passos que deve ser seguida para a realização correta do mapeamento.

Além disso, você aprendeu que os dicionários de dados são úteis para documentar seu Modelo Relacional.



# Obrigado!



paulo.perris@garanhuns.ifpe.edu.br

