Banco de Dados

Aula 05 - Transformações ER para MR e dicionário de dados

Apresentação

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.

Vídeo 01 - Apresentação

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?

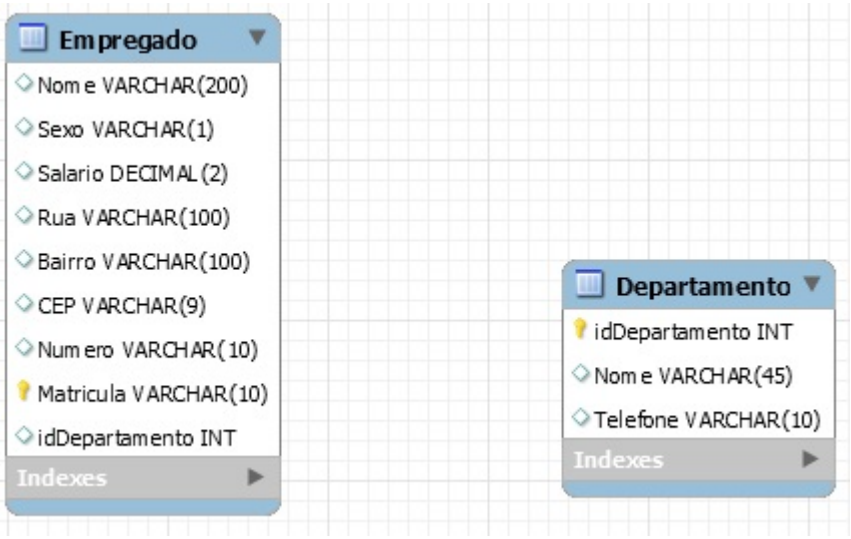
Sexto passo: mapear relacionamento um-para-um

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. Note que este exemplo já foi mostrado na Aula 02 (precisamente na Figura 8).

Figura 01 - Relacionamento um-para-um no diagrama ER.

Para mapear este relacionamento para o Modelo Relacional, devemos inserir em uma tabela uma chave estrangeira que referencie 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.

Figura 02 - Tabelas sem relacionamento um-para-um.



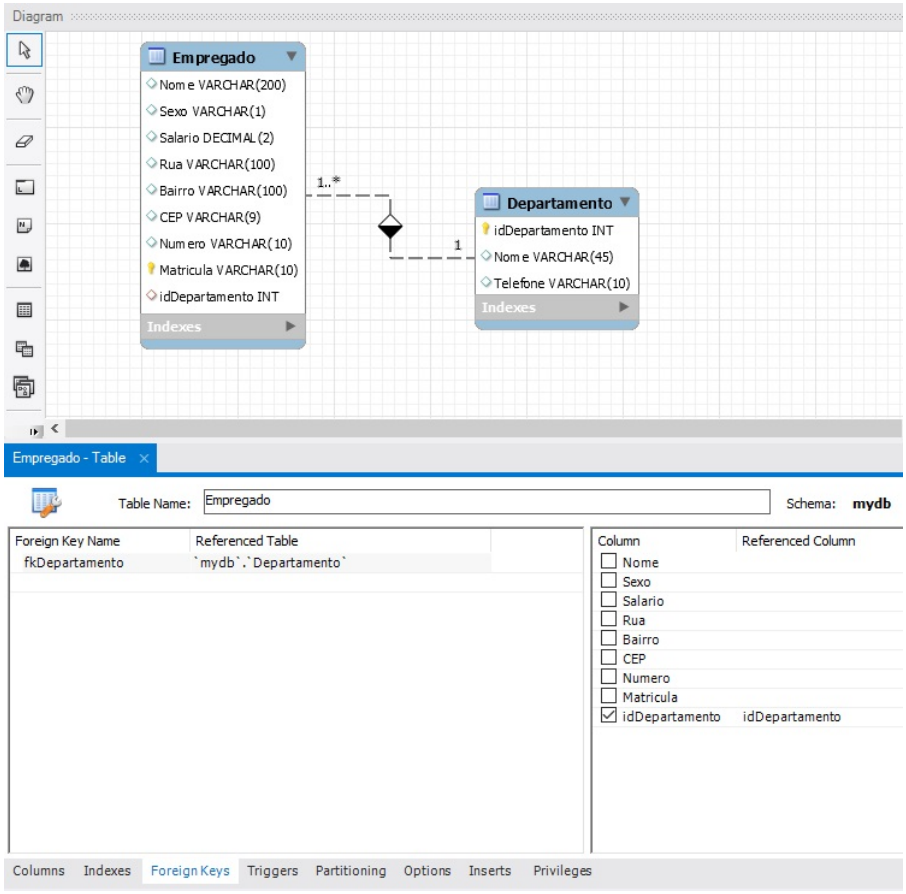
Agora, para definir o relacionamento no *MySQL Workbench*, você deve seguir os seguintes subpassos:

1. 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**.

Figura 03 - Definindo relacionamento um-para-um.

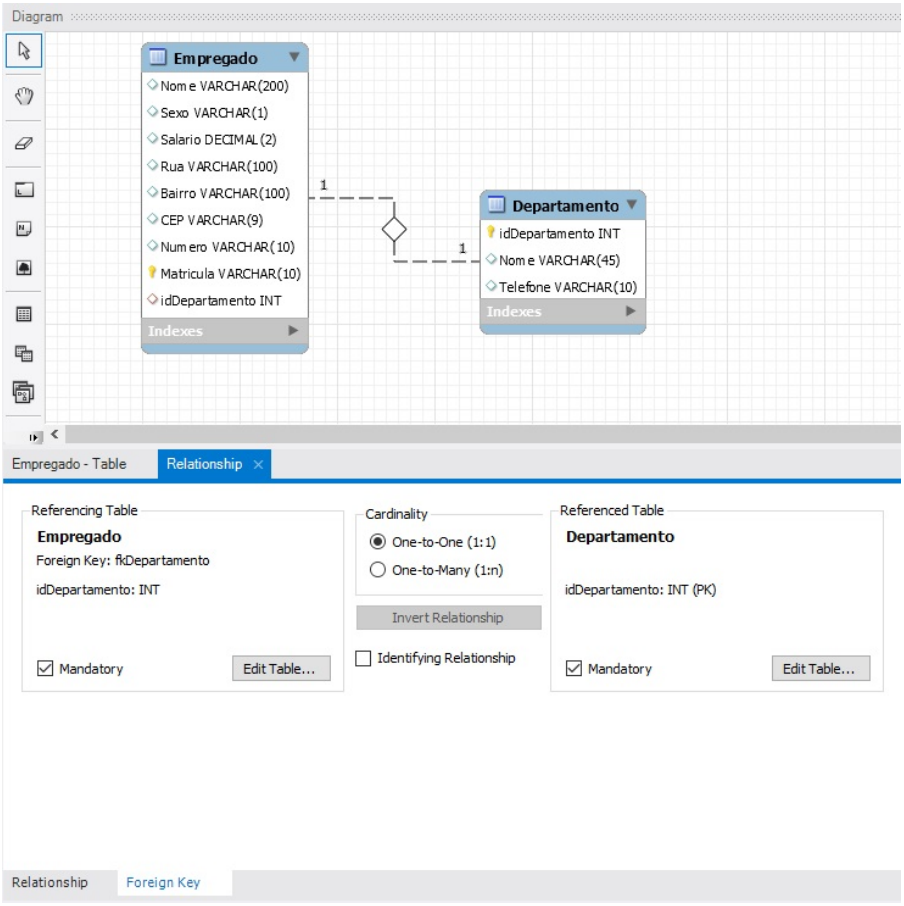
2. Definir uma chave estrangeira com o atributo idDepartamento para referenciar a chave primária da tabela Departamento. Lembra como fazer essa referência entre chaves? Caso tenha alguma dificuldade, volte à Aula 03. Ao final deste subpasso, você deve obter algo parecido com a **Figura 4**.

Figura 04 - Definindo relacionamento um-para-um.



3. 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**.

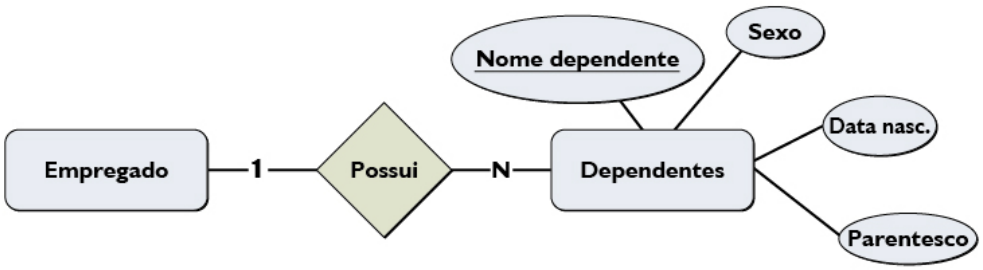
Figura 05 - Definindo a cardinalidade do relacionamento.



Sétimo passo: mapear relacionamento um-para-muitos

Você se lembra da nossa Aula 02? 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.

Figura 06 - Relacionamento um-para-muitos no diagrama ER.



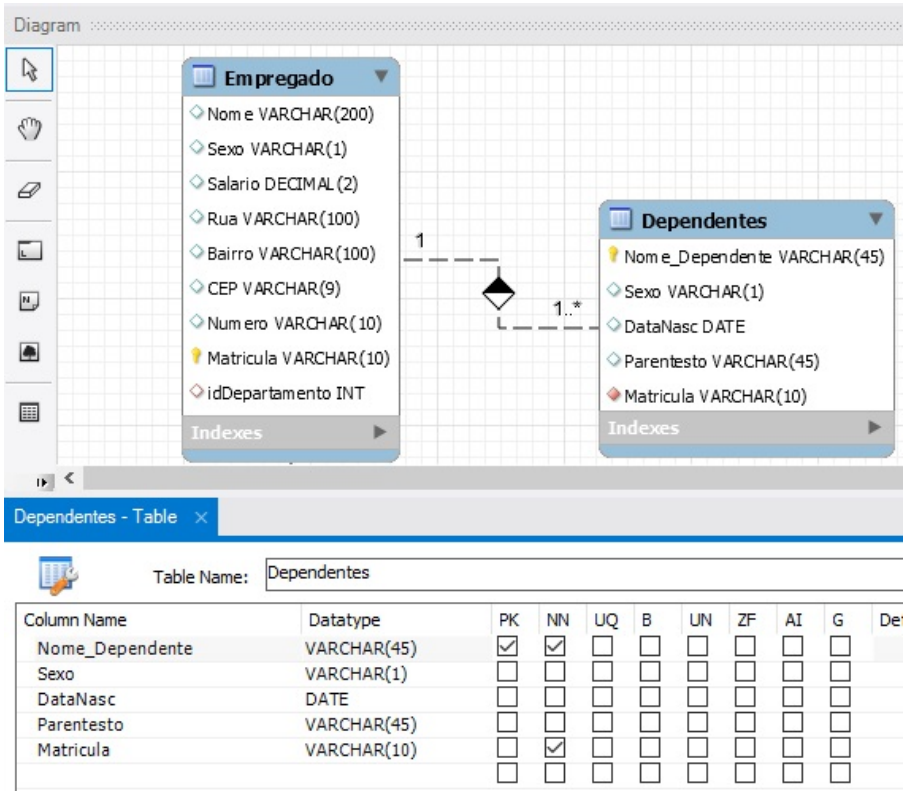
Agora veremos como mapear tal relacionamento para o Modelo Relacional. Para definir o relacionamento no *MySQL Workbench*, você deve primeiro mapear as entidades Empregado e Dependentes para o Modelo Relacional, como já vimos anteriormente. Depois você deve realizar os seguintes subpassos:

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**.

Figura 07 - Definindo relacionamento um-para-muitos.

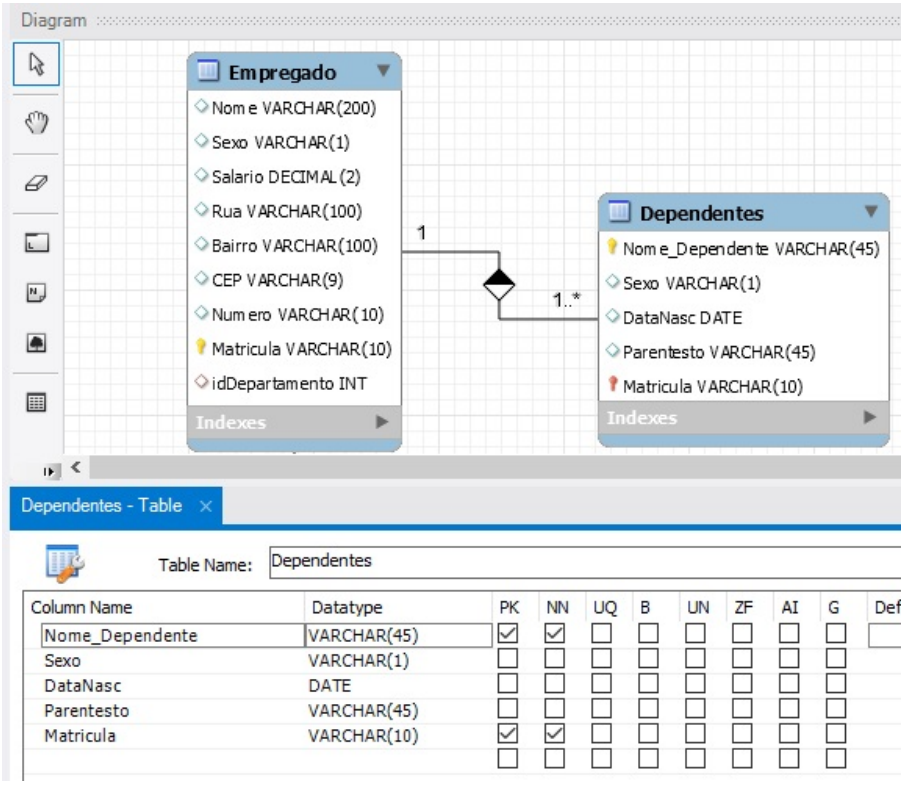
2. Definir uma chave estrangeira com o atributo Matricula para referenciar a chave primária da tabela Empregado. Ao final deste subpasso, você deve obter algo parecido com a **Figura 8**.

Figura 08 - Definindo relacionamento um-para-muitos.



3. 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**.

Figura 09 - Definindo a chave primária de um relacionamento um-para-muitos.

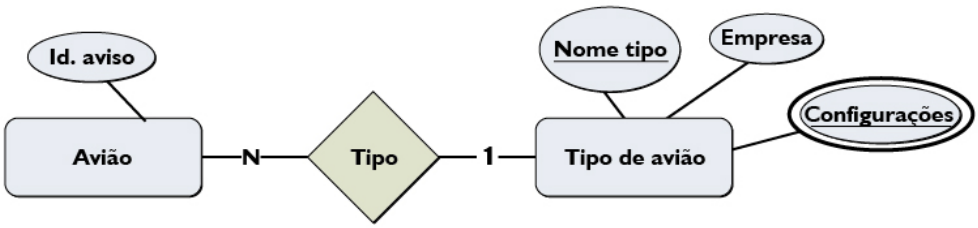


Note que agora dois dependentes podem ter o mesmo nome, desde que eles estejam vinculados a empregados diferentes.

Vídeo 02 - Conversão ER para MR em um relacionamento 1:N

Atividade 01

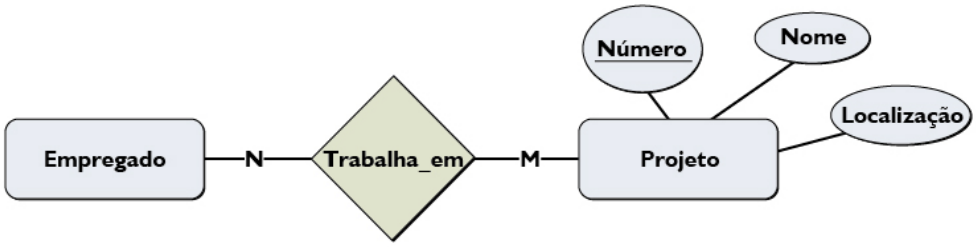
1. Repita os passos até aqui estudados para definir o mapeamento do relacionamento entre Avião e Tipo Avião.



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.

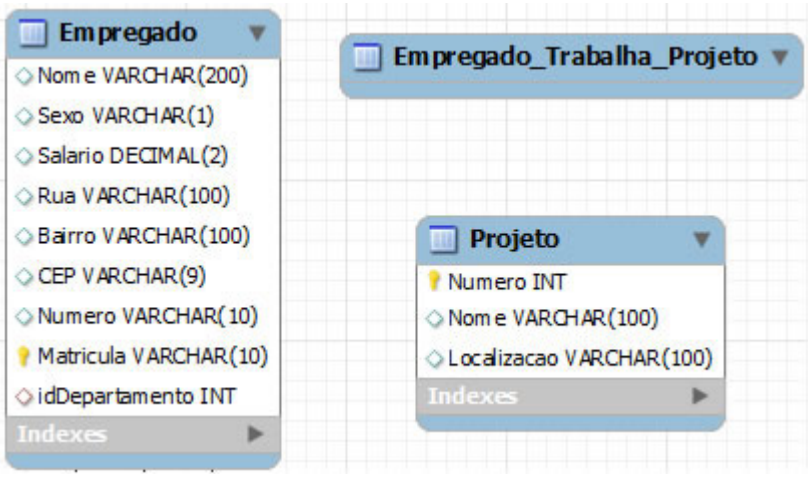
Figura 10 - Relacionamento muitos-para-muitos no diagrama ER.



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. Depois você deve realizar os seguintes subpassos:

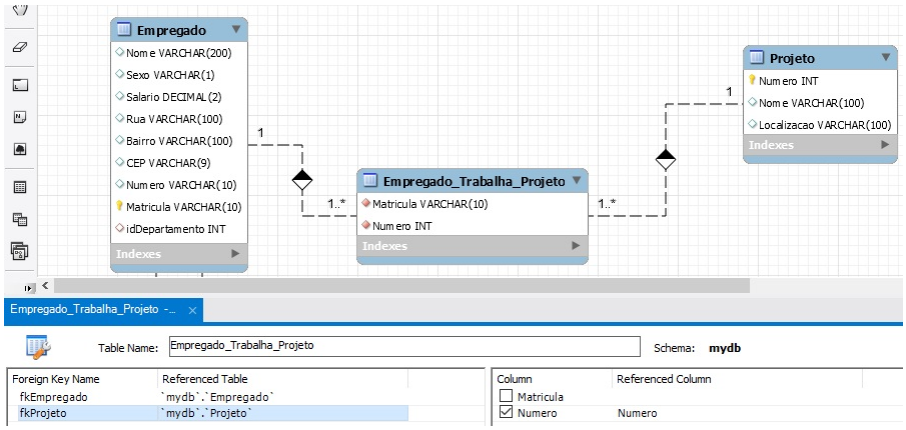
1. Criar uma nova tabela para representar o relacionamento muitos-para-muitos. No nosso exemplo, você poderia criar uma tabela com o nome Empregado_Trabalha_Projeto para representar o relacionamento no modelo ER. Ao final deste subpasso, você deve obter algo parecido com a **Figura 11**.

Figura 11 - Tabela criada para representar um relacionamento muitos-para-muitos.



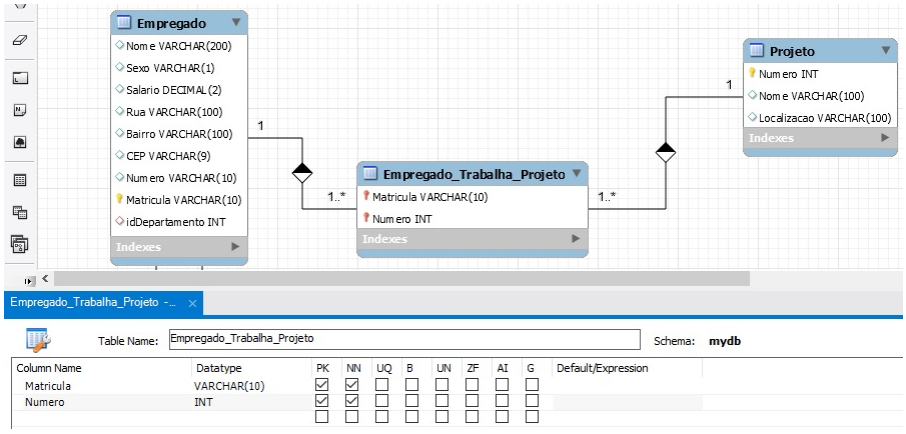
2. Inserir, como chave estrangeira na tabela recém-criada, as chaves primárias das entidades participantes. No caso do exemplo, você deve inserir como chave estrangeira os atributos Matrícula e Número, que são as chaves primárias das tabelas Empregado e Projeto, respectivamente. Ao final deste subpasso,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

Figura 12 - Criando chave estrangeira na tabela Empregado_Trabalha_Projeto.



3. 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**.

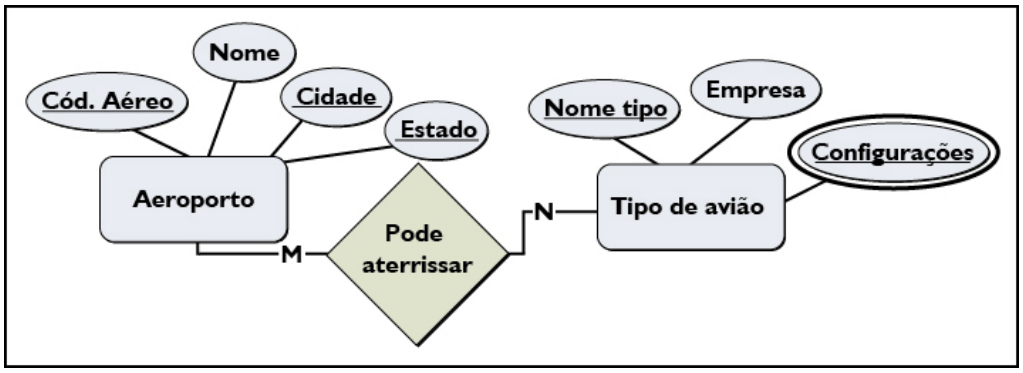
Figura 13 - Definindo a chave primária da tabela Empregado_Trabalha_Projeto.



Vídeo 03 - Conversão ER para MR em um relacionamento N:M

Atividade 02

1. Siga os passos até aqui estudados para definir o mapeamento do relacionamento entre Aeroporto e Tipo Avião.



Dicionário de dados

Nas Aulas 03 e 04, 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 ao lado.

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

Vídeo 04 - Dicionário de Dados

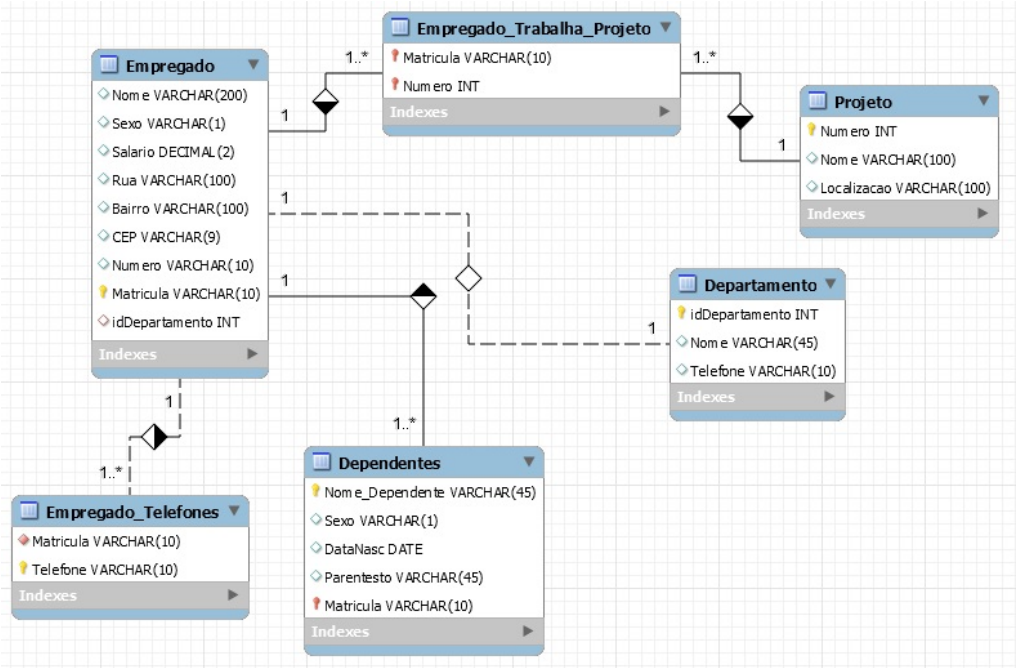
Atividade 03

1. Documente o Modelo Relacional da Empresa mostrado na **Figura 14** através de um dicionário de dados.

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**.

Figura 14 - Diagrama final da aula no *MySQL Workbench*.

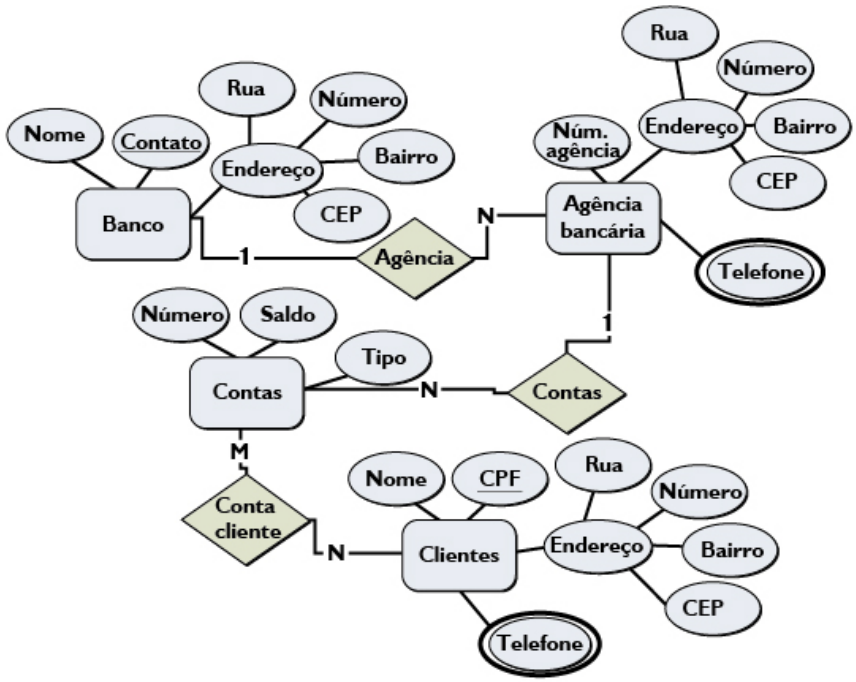


Resumo

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.

Autoavaliação

1. Faça o mapeamento do modelo ER do sistema bancário descrito a seguir para o Modelo Relacional, utilizando a ferramenta *MySQL Workbench*.



2. Elabore o dicionário de dados para o modelo bancário mapeado no item 1.

Referências

DATE, C. J. **Introduction to database systems.** 7th ed. Boston: Addison Wesley Longman, 1999.

_____. **Introdução a sistemas de banco de dados.** Rio de Janeiro: Campus, 2000.

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de banco de dados.** 4. ed. São Paulo: Addison Wesley, 2005.

HEUSER, C. A. **Projeto de Banco de Dados.** 6. ed. São Paulo: Editora Bookman, 2009.

SANCHES, Andre Rodrigo. **Modelos entidade relacionamento.** Disponível em: <http://www.ime.usp.br/~andrers/aulas/bd2005-1/aula7.html>. Acesso em: 27 ago. 2012.