

C07 - BD

Arthur Openheimer

Comando CREATE

```
CREATE TABLE IF NOT EXISTS empresa(  
    matricula INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(45),  
    cpf VARCHAR(45),  
    cargo VARCHAR(45),  
    nProjetos INT  
);
```

Comando CREATE

```
CREATE TABLE IF NOT EXISTS empresa(  
    matricula INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(45),  
    cpf VARCHAR(45),  
    cargo VARCHAR(45),  
    nProjetos INT  
);
```

matricula INT NOT NULL AUTO_INCREMENT PRIMARY KEY,

Tipo do campo

Não pode ser nulo
(Por padrão todos os
campos podem ser nulos
se não tiverem esse
modificador)

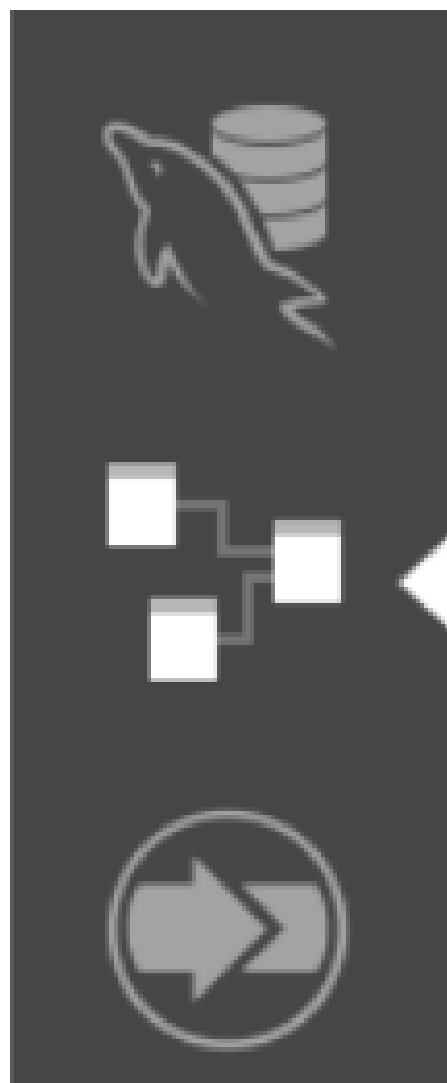
Para campos numéricos, faz com que não
seja necessário preencher esse campo no
comando INSERT, o próprio modificador
busca o último número cadastrado,
incrementa seu valor em 1 e preenche o
campo

Indica que o campo é a chave
primária, ou seja, deve ser único

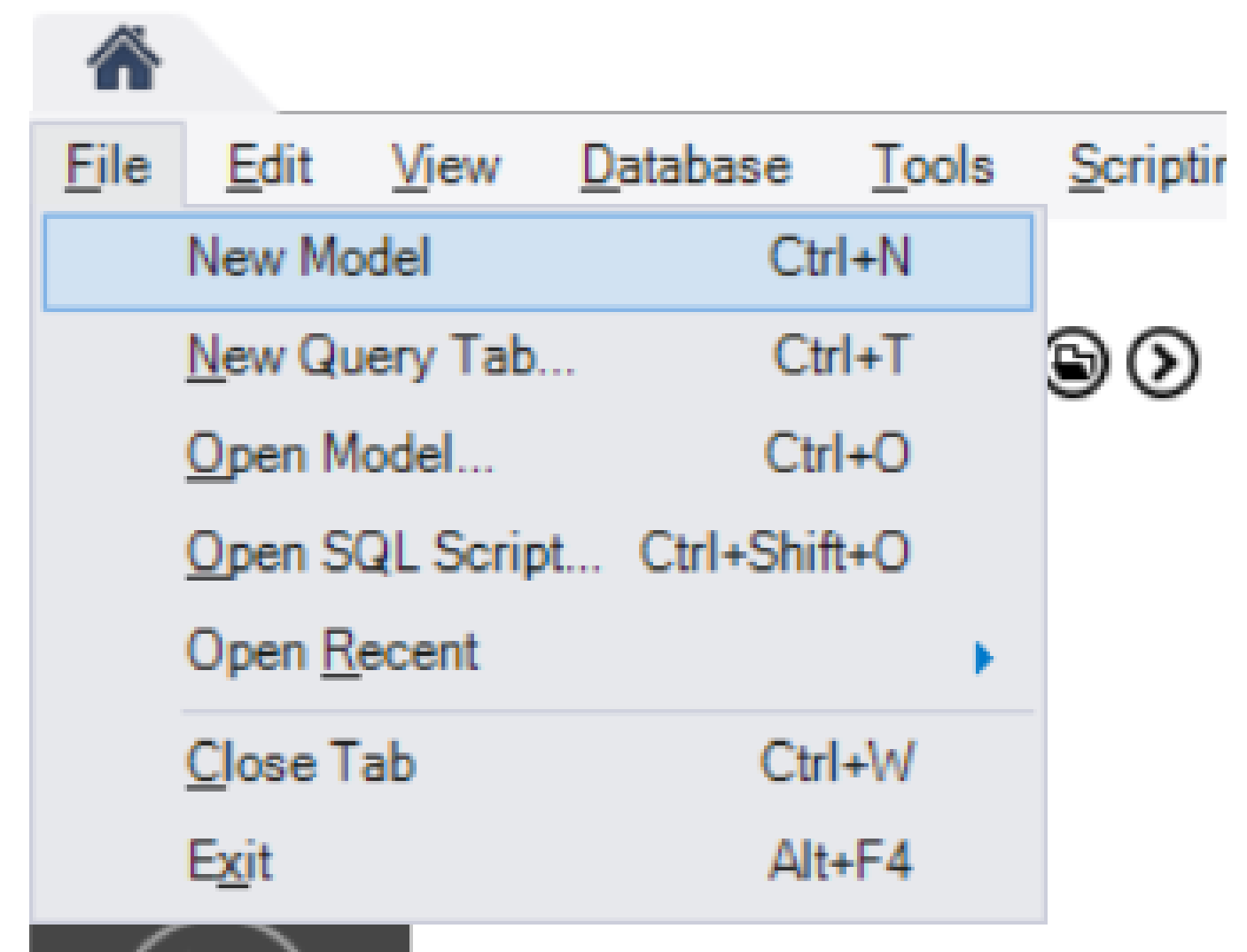
Criando tabelas com modelos

Existem duas formas de se criar um novo Modelo:

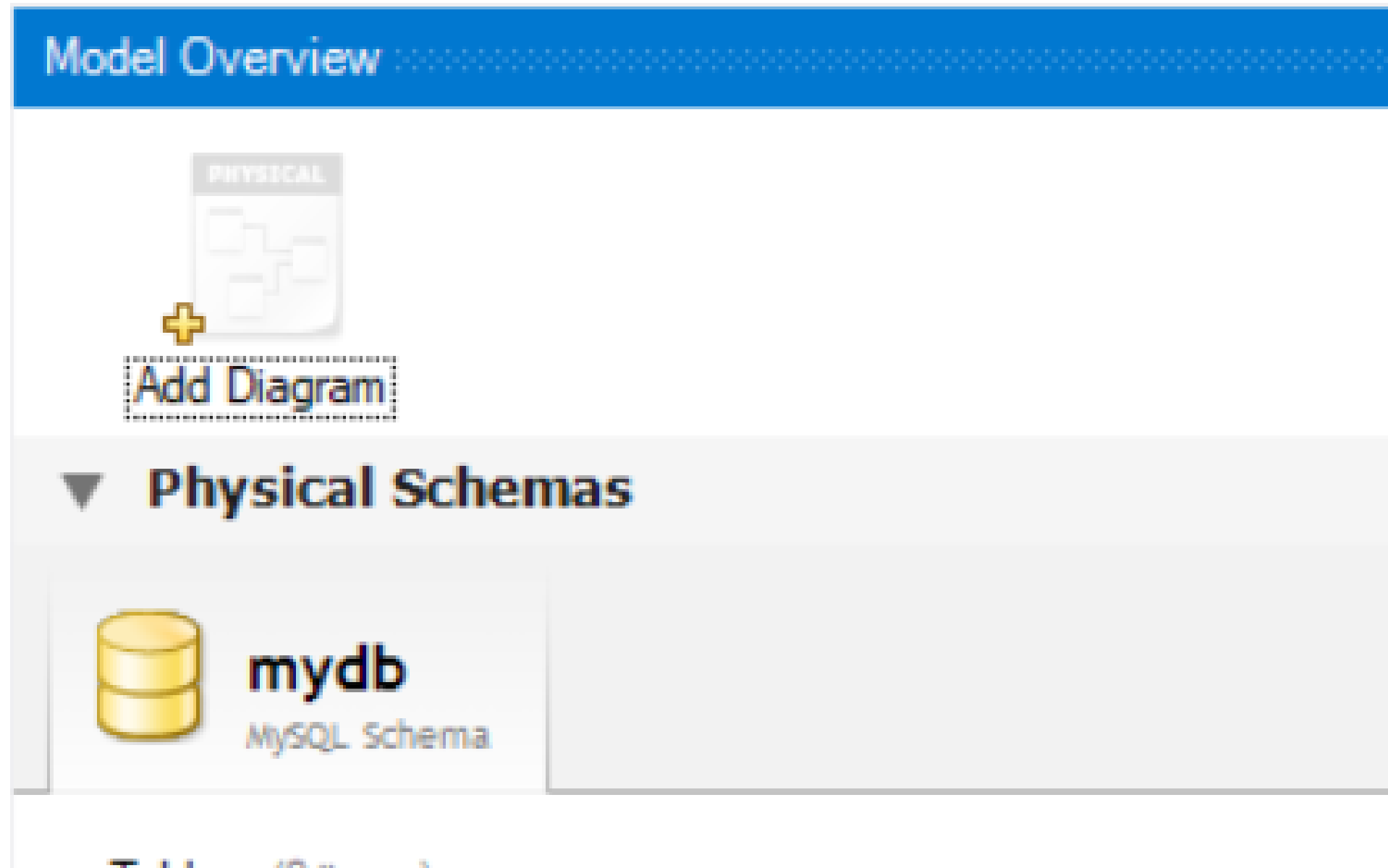
- Clicar em + na aba Models
- Clicar em File -> New Model



Models ⊕ 📄 ➡




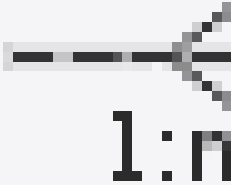


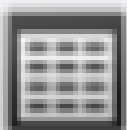


Criando tabelas com modelos

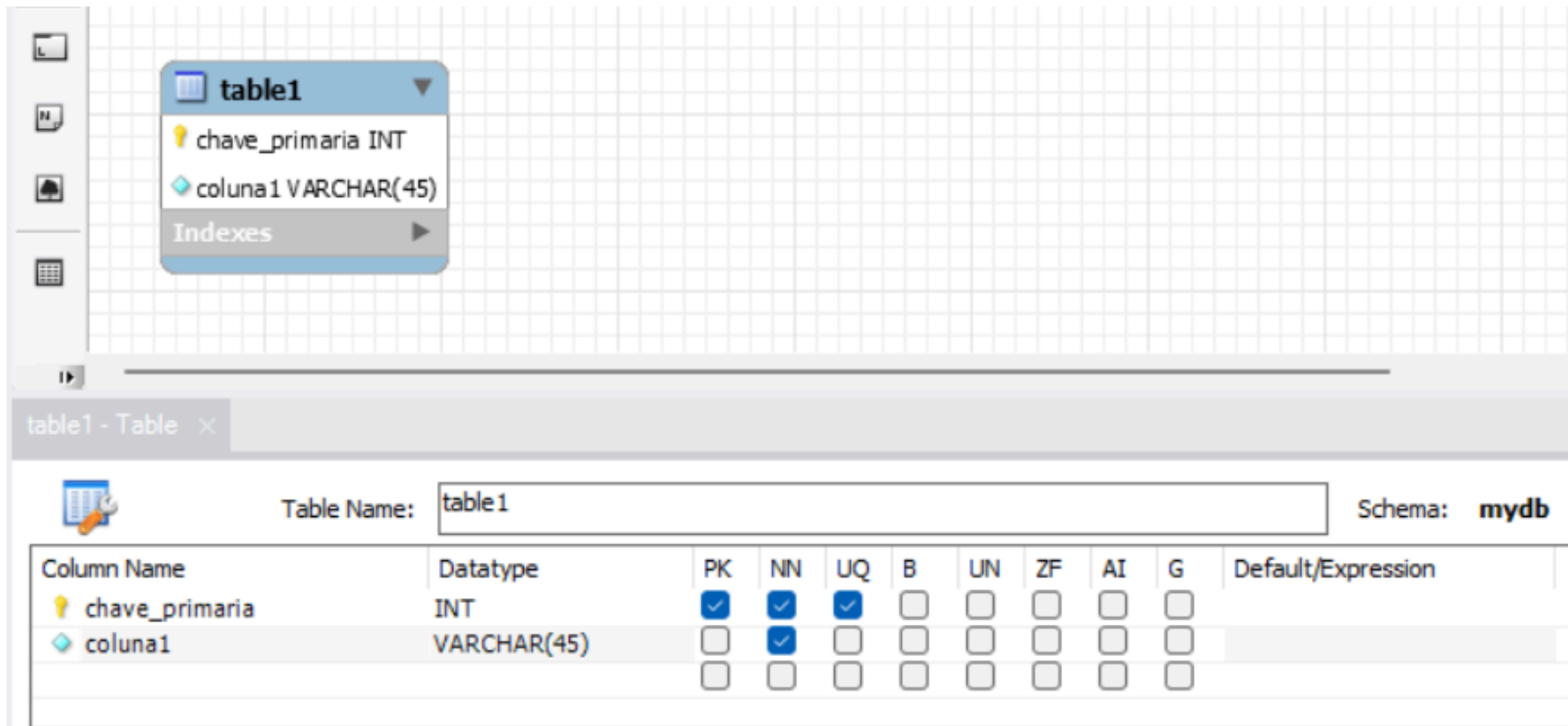


Basta clicar duas vezes em “Add Diagram” para criar um diagrama de entidade-relacionamento

Criando tabelas com modelos



	Cursor para seleção		Botão para criar um relacionamento 1:1
	Cursor para mover o diagrama		Botão para criar um relacionamento 1:N (A primeira tabela selecionada deve ser a tabela N)
	Borracha para apagar dados		Botão para criar um relacionamento N:N
	Botão para adicionar uma tabela ao diagrama		

Criando tabelas com modelos



Clique duas vezes na tabela para abrir o menu de colunas

Criando tabelas com modelos

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 chave_primaria	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
 coluna1	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

↓

Clique duas vezes no espaço em branco para adicionar uma nova coluna

↓

Alterar o tipo do dado armazenado na coluna

↓

Opções da coluna:

- PK - Chave primária
- NN - Não nulo
- UQ - Valor único
- B - Valor booleano
- UN - Sem parte negativa
- ZF - Preencher com zero
- AI - Auto incremento
- G - Gerada(não entrada)

Relacionamento 1:1

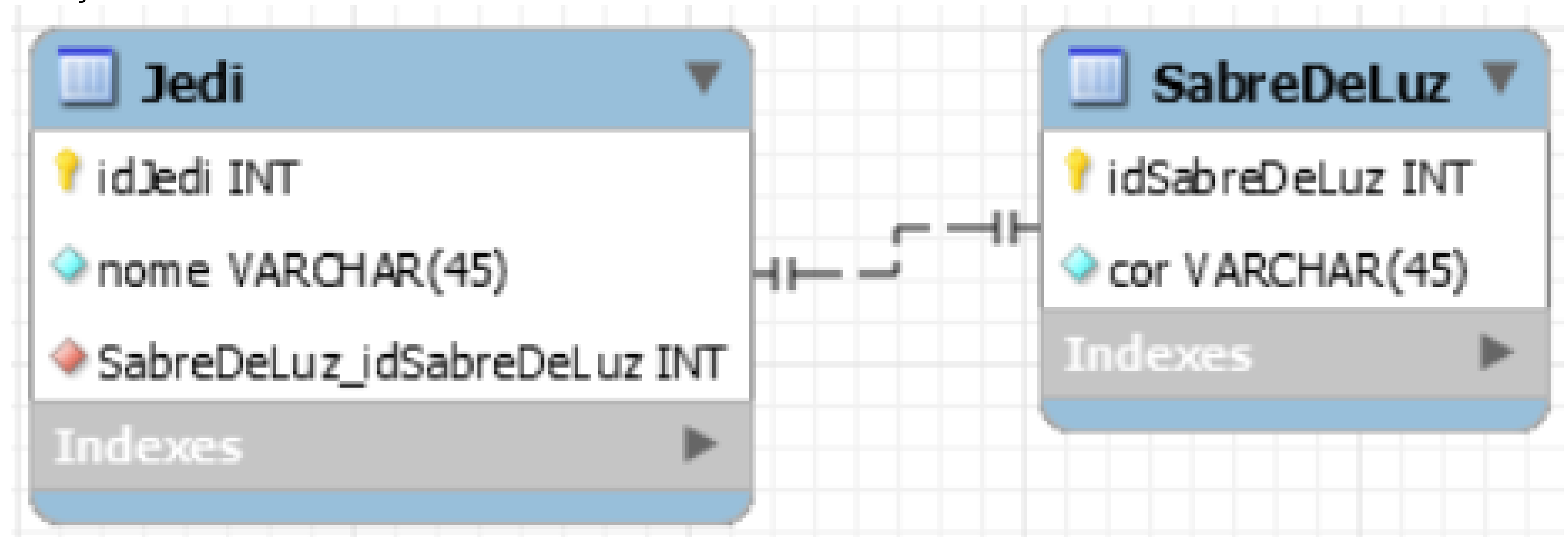
Em Star Wars, um Jedi só pode possuir um único sabre de luz, e um sabre de luz pode pertencer a apenas um único Jedi

Como relacioná-los em um banco de dados?



Relacionamento 1:1

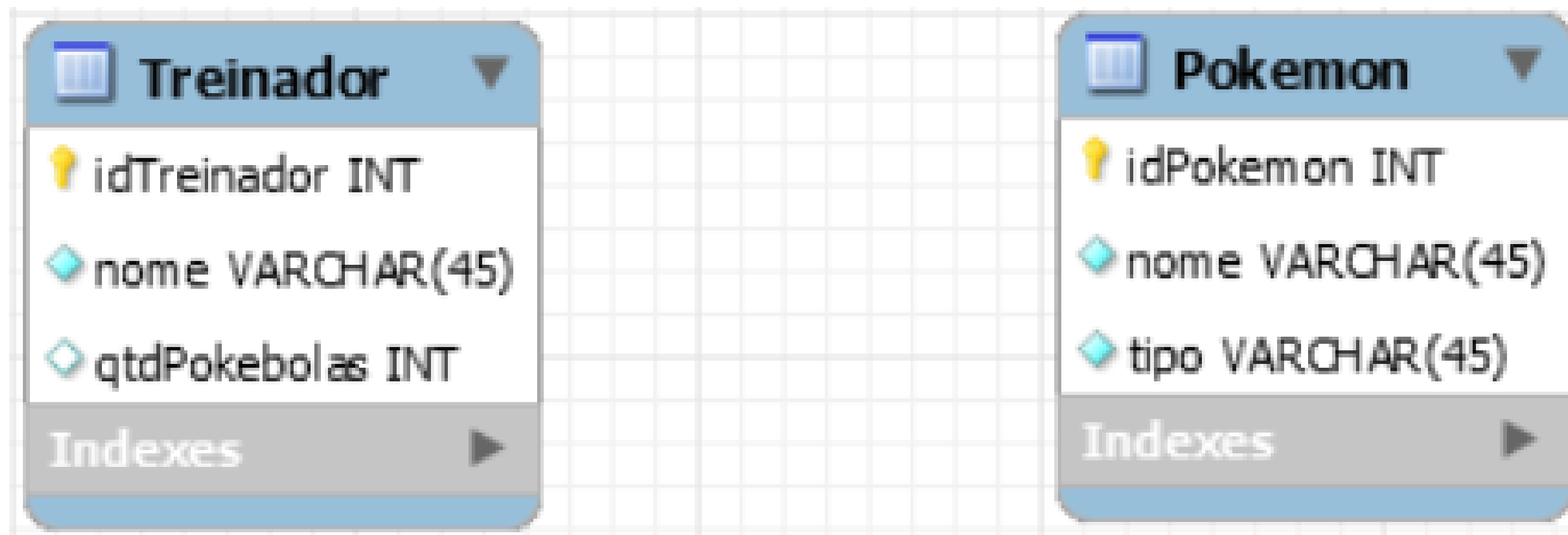
Um relacionamento 1:1 adiciona uma nova coluna em uma tabela que armazena a chave primária de outra tabela como uma chave estrangeira(foreign key) que é usada para fazer a relação entre tabelas



Relacionamento 1:N

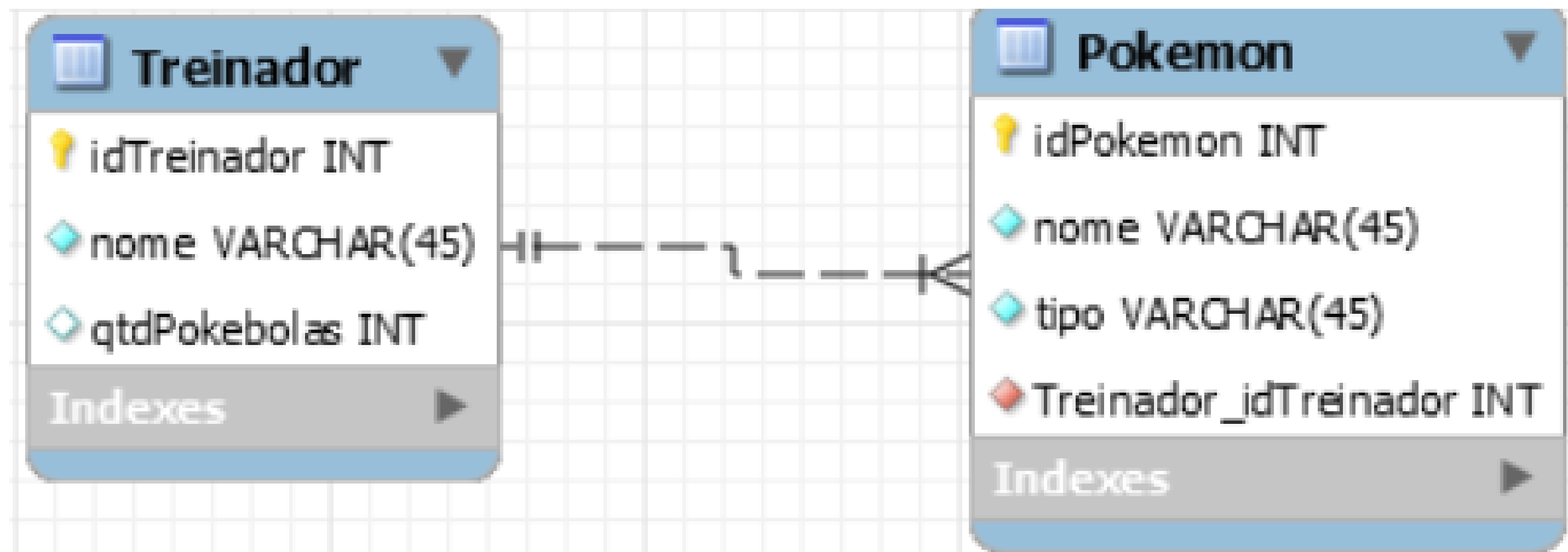
Em Pokémon, um treinador pode capturar N Pokémon, mas um Pokémon só pode pertencer a um único treinador

Como relacioná-los em um banco de dados?



Relacionamento 1:N

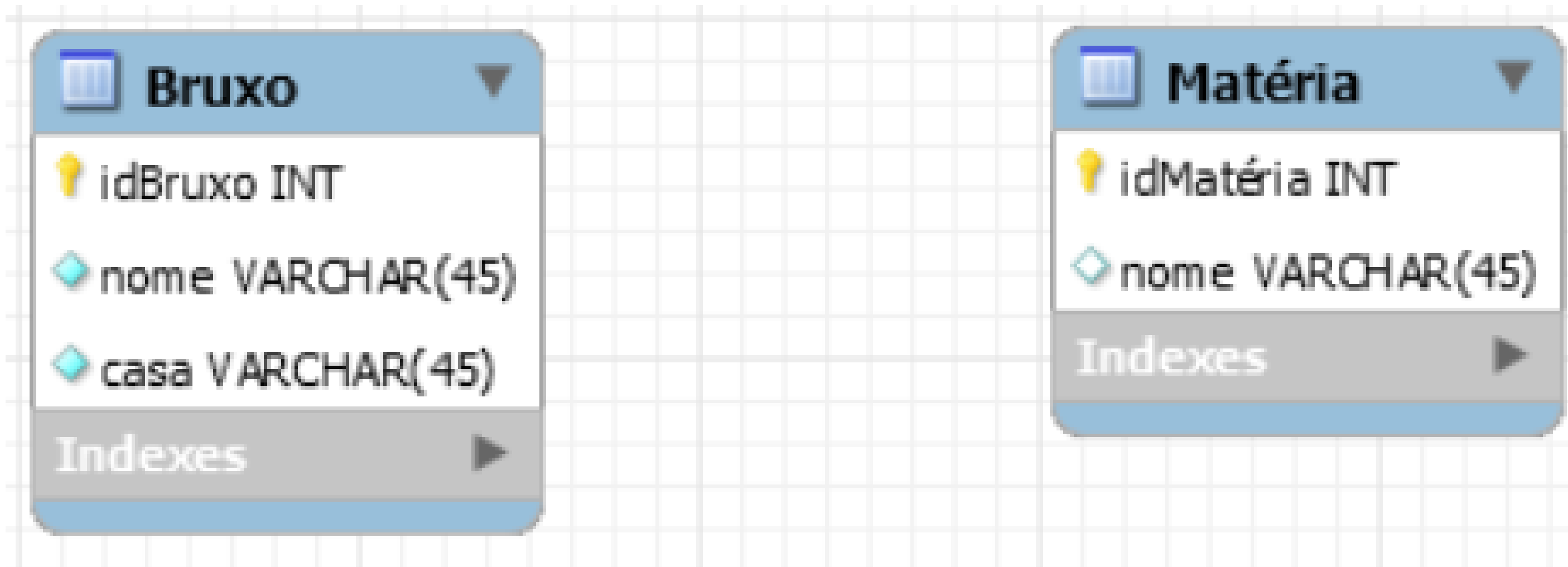
Um relacionamento 1:N adiciona uma nova coluna na tabela “N”, que armazena a chave primária da tabela “1” como uma chave estrangeira(foreign key) que é usada para fazer a relação entre tabelas



Relacionamento N:N

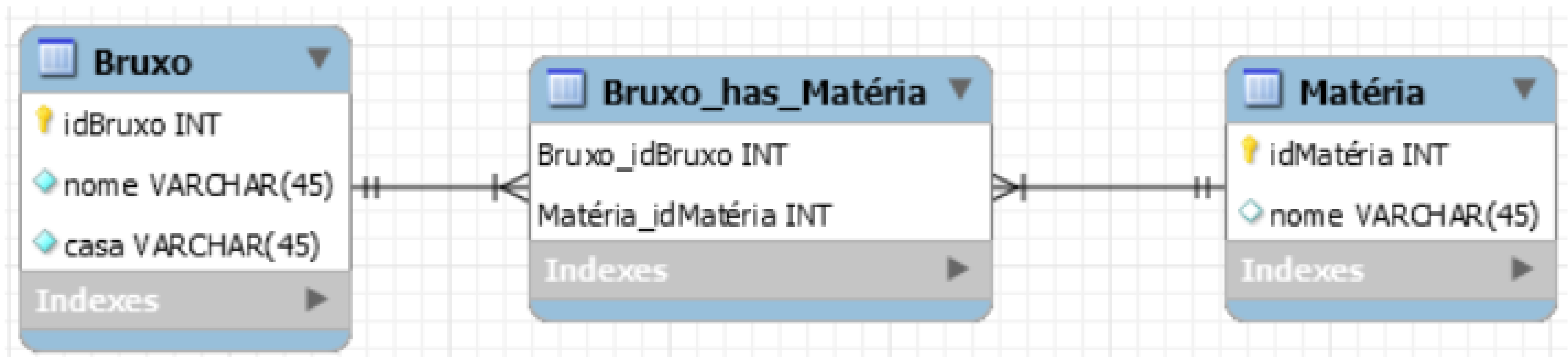
Em Hogwarts, um bruxo precisa estudar N matérias para passar de ano, e uma matéria pode ter N bruxos inscritos

Como relacioná-los em um banco de dados?



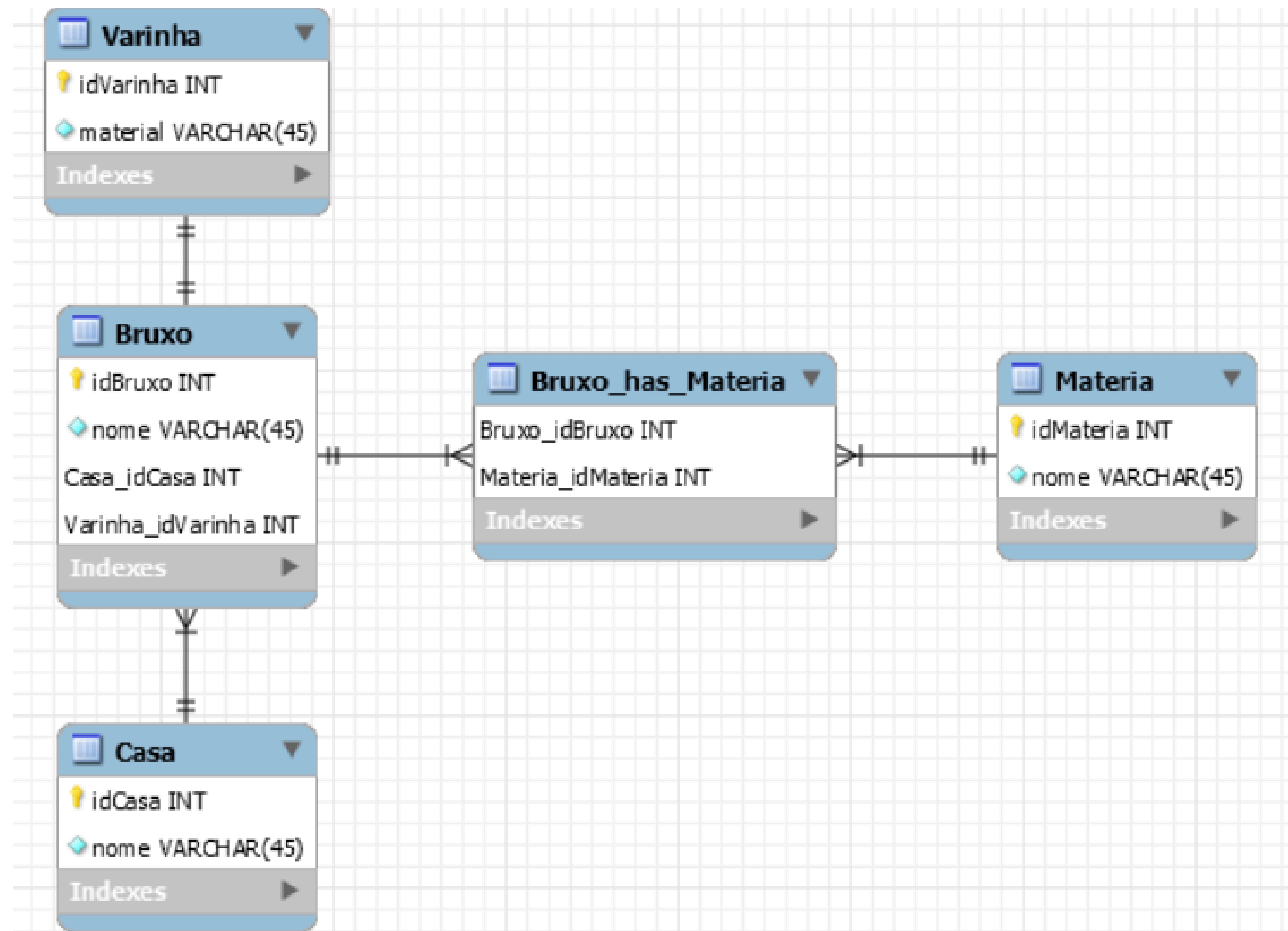
Relacionamento N:N

Um relacionamento N:N cria uma tabela auxiliar, pois não seria possível criar uma nova coluna para vários valores. A tabela consiste em armazenar as chaves primárias de ambas as tabelas como chaves estrangeiras, criando a relação entre elas



Relacionamentos

Ainda no exemplo de Harry Potter, um esquema utilizando todos os tipos de relacionamentos



Exportando model para código

- Uma das ferramentas mais úteis da ferramenta da modelagem de banco de dados é a exportação do modelo para um código SQL
- Primeiro, salve o modelo clicando em File->Save Model
- Depois, para criar o código SQL basta clicar em Database->Forward Engineer, selecionar uma conexão de banco de dados e clicar em “next” até aparecer o código SQL na janela
- No código SQL, deve-se apagar as linhas com “SET” e “ENGINE”, salvar o arquivo e clicar em “next”
- Agora basta acessar a conexão, ir em Files->Open SQL Script

Chaves estrangeiras

```
CREATE TABLE Bruxo (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  nome VARCHAR(255) NOT NULL,  
  Casa_idCasa INT NOT NULL,  
  Varinha_idVarinha INT NOT NULL,  
  CONSTRAINT fk_Casa  
  FOREIGN KEY (Casa_idCasa)  
  REFERENCES Casa(idCasa),  
  CONSTRAINT fk_Varinha  
  FOREIGN KEY (Varinha_idVarinha)  
  REFERENCES Varinha(idVarinha)  
  ON UPDATE CASCADE  
  ON DELETE CASCADE  
);
```

- CONSTRAINT - Bloco de comandos para configurar a chave estrangeira. Para cada chave estrangeira da tabela um bloco CONSTRAINT deve ser feito
- ON UPDATE e ON DELETE - Definem o que fazer estrangeira. ao se excluir um registro em outra tabela relacionado pela chave estrangeira
- CASCADE - A alteração na tabela inicial se estende ao registro da tabela com a chave estrangeira
- FOREIGN KEY - Comando que mostra qual coluna é uma chave estrangeira
- REFERENCES - Referencia a qual tabela a chave estrangeira está relacionada

Exercício

- Um empresário deseja se envolver com o mercado de E-Commerce. Ele precisa modelar o banco de dados do seu futuro marketplace (shopping virtual), a SHOP-E. As informações necessárias para a modelagem do banco de dados estão descritas a seguir:
- Deve-se guardar estas informações de um produto: cada produto deve ter uma identificação única, um nome, uma marca, um tipo (smartphones, periféricos, eletrônicos em geral, etc.), o seu preço e a quantidade disponível no estoque.
- Deve-se guardar estas informações de uma loja: cada loja deve ter uma identificação única, um nome, um endereço, um telefone e a informação opcional de e-mail. Um produto é vendido por várias lojas diferentes e uma única loja vende um ou mais produtos distintos.
- Deve-se guardar estas informações de um cliente: cada cliente deve ter um CPF para identificação única, um nome, o seu CEP (apenas números), um e-mail, uma senha única (aceita letras, números e caracteres especiais), e a informação opcional de telefone.
- Um cliente pode comprar vários produtos assim como um produto é comprado por muitos clientes. Toda compra possui um valor total (aceita centavos), um tipo de pagamento (cartão ou boleto, por exemplo) e um código de rastreamento único.
- Deve-se guardar estas informações do correio: o correio deve possuir um código de identificação único, o prazo de entrega dos produtos (quantidade de dias) e o valor do frete calculado (aceita centavos). Um único correio é responsável pela entrega de todas as compras e cada compra é entregue por apenas esse correio.
- Após a entrega, um cliente pode avaliar diversos produtos e cada produto pode ser avaliado por vários clientes. Toda avaliação feita pelo cliente deve possuir uma nota de um a cinco. O cliente tem a opção de digitar um texto para esclarecer a nota destinada ao produto.