

# Banco de Dados – C07

Felipe Tagawa Reis

Comando CREATE  
Models

CREATE DATABASE BD;

# Comando CREATE

CREATE USER 'Monitor' IDENTIFIED BY 'senha';

Cláusula

```
CREATE TABLE Cliente (  
    codigo INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(45) NOT NULL,  
    telefone VARCHAR(15) NOT NULL  
);
```

codigo INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

Nome do  
Atributo

Tipo do  
Atributo

Indica que o  
Atributo não  
pode ser  
nulo (NOT  
NULL)

Para Atributos  
numéricos, esse  
modificador incrementa  
em 1 no último inserido,  
sem a necessidade de  
inserir manualmente

Indica que este  
é um Atributo  
chave para a  
tabela em  
questão

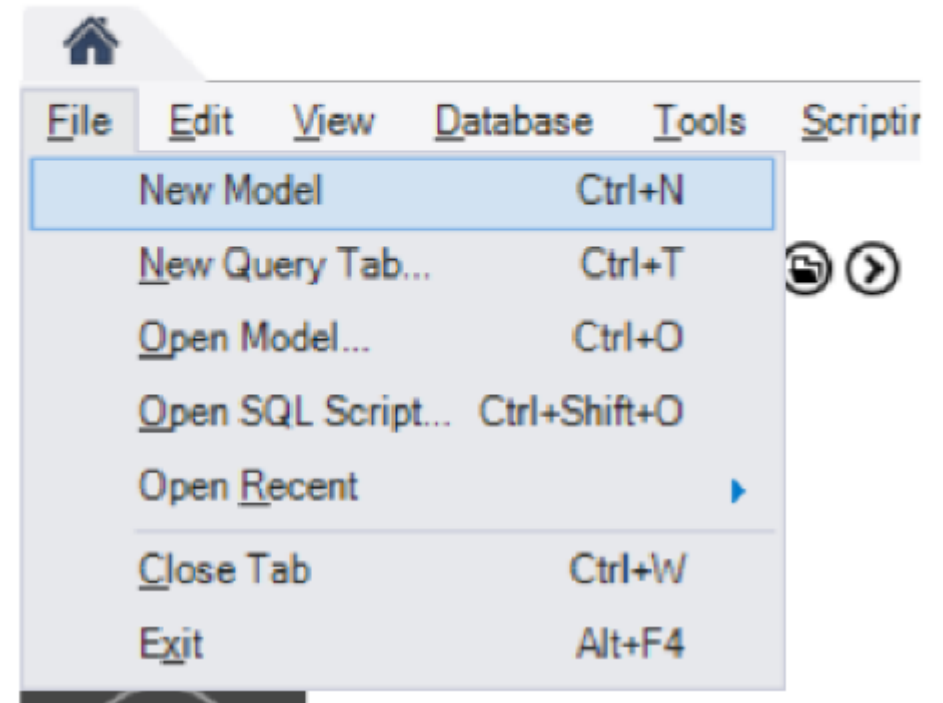
# Criando Models

Existem duas formas de se criar um novo Modelo:

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

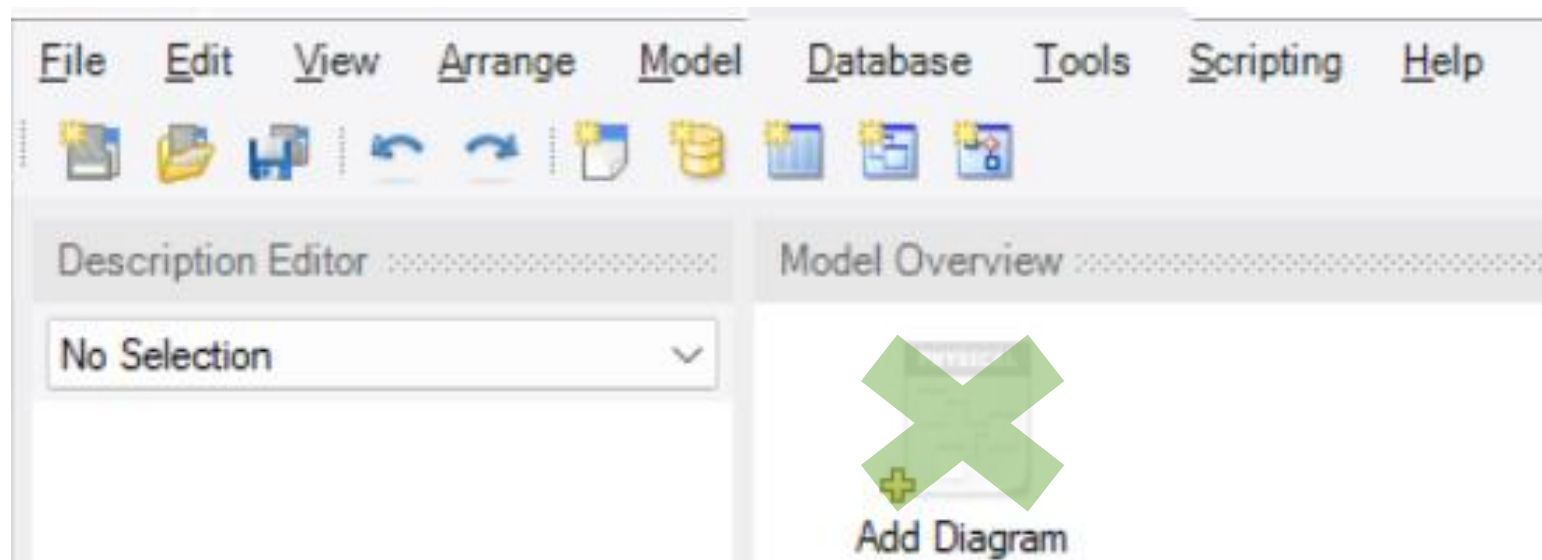


Models   



# Criando Models

Basta clicar duas vezes em “Add Diagram”



# Criando Models



Cursor para seleção



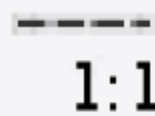
Cursor para mover o diagrama



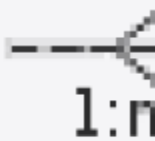
Borracha para apagar dados



Botão para adicionar uma  
tabela ao diagrama



Botão para criar um relacionamento 1:1





Botão para criar um relacionamento 1:N  
(A primeira tabela selecionada deve ser  
a tabela N)



Botão para criar um relacionamento N:N

# Criando Models

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
 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)

# Relacionamentos

**Conexões lógicas entre entidades** que representam como os dados em diferentes tabelas estão associados uns aos outros.

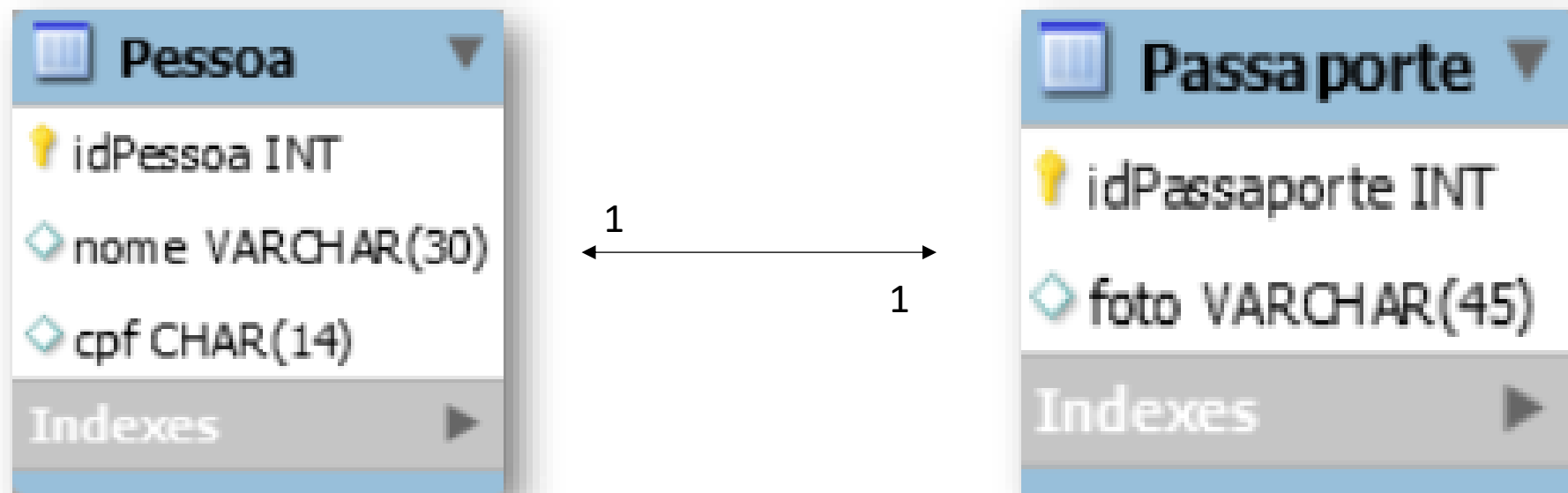
No modelo relacional, **relacionamentos** ajudam a manter a **integridade referencial**, evitando dados duplicados e inconsistentes.

Classificação:

- 1:1
- 1:N
- N:M

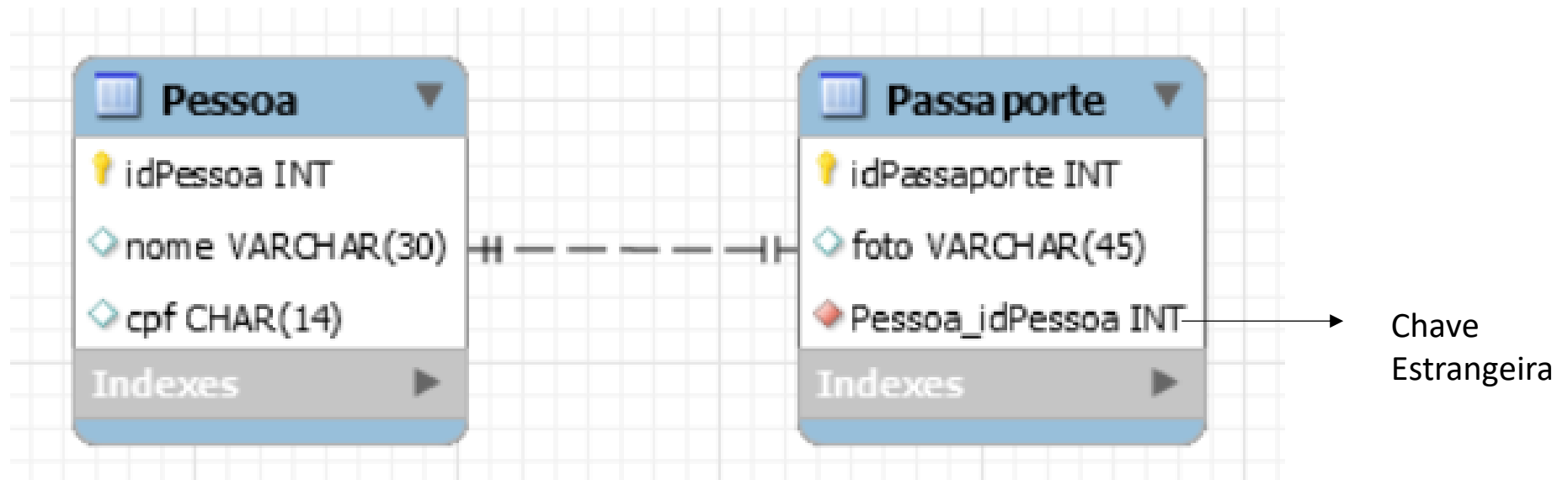
# Relacionamento 1:1

Em questão de documentos, qual a relação de uma pessoa com um passaporte?  
Uma pessoa tem 1 passaporte exclusivo e 1 passaporte pertence a uma única pessoa.





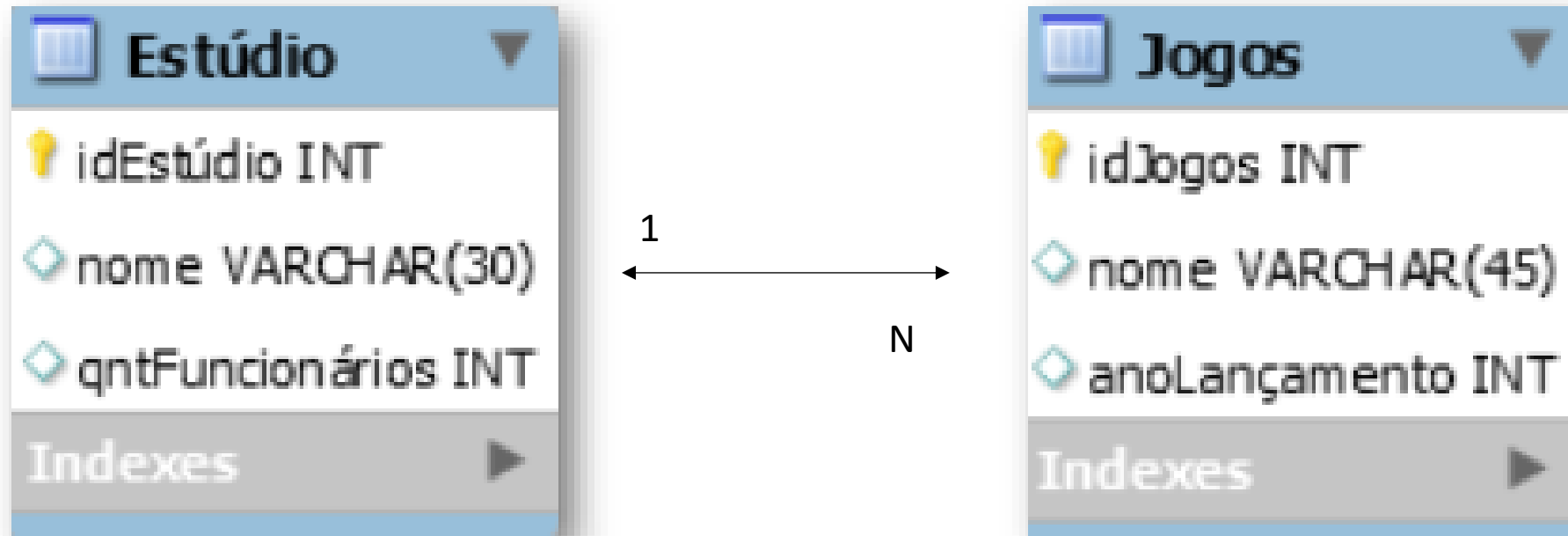
# Relacionamento 1:1



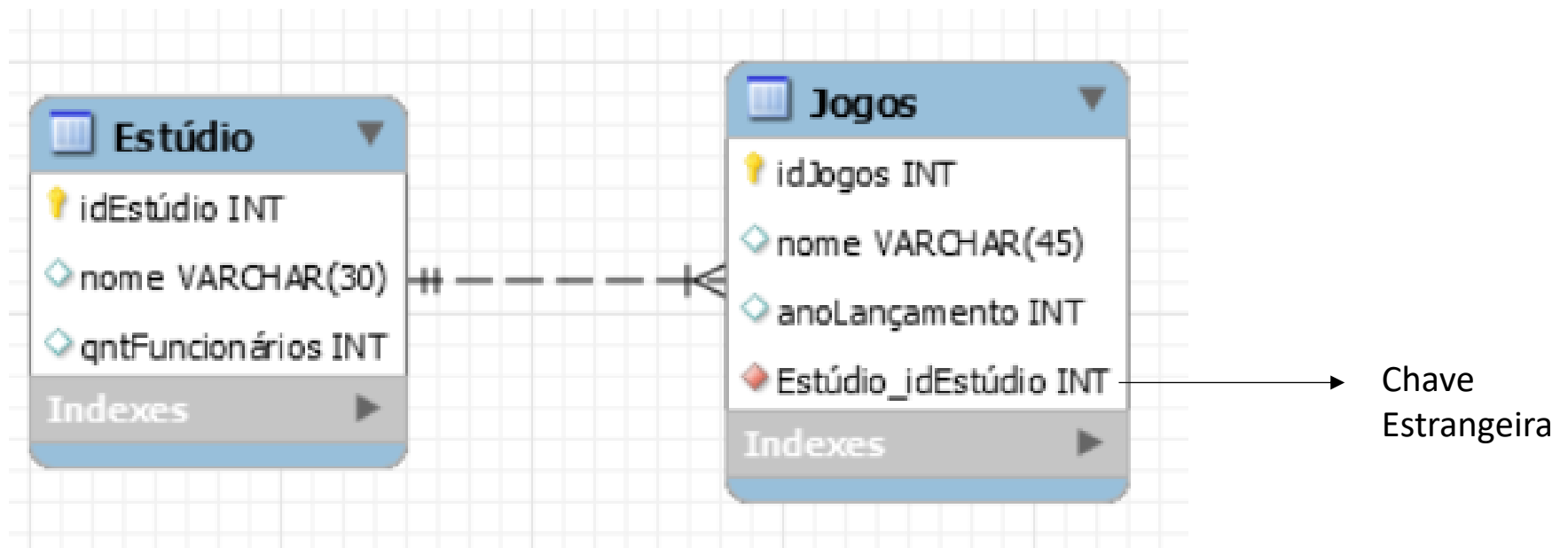
Ao associar duas entidades(tabelas), é gerada em uma delas uma chave estrangeira(FK) que é usada pra conectá-las. Essa *Foreign Key* pode ser colocada em qualquer um dos lados em uma relação 1:1, mas em casos em que há entidades fracas, a chave deve ser alocada no lado da mesma.

# Relacionamento 1:N

Em uma relação entre Estúdio e Jogos, 1 Estúdio pode desenvolver N jogos. Mas cada jogo está associado a um estúdio específico.



# Relacionamento 1:N

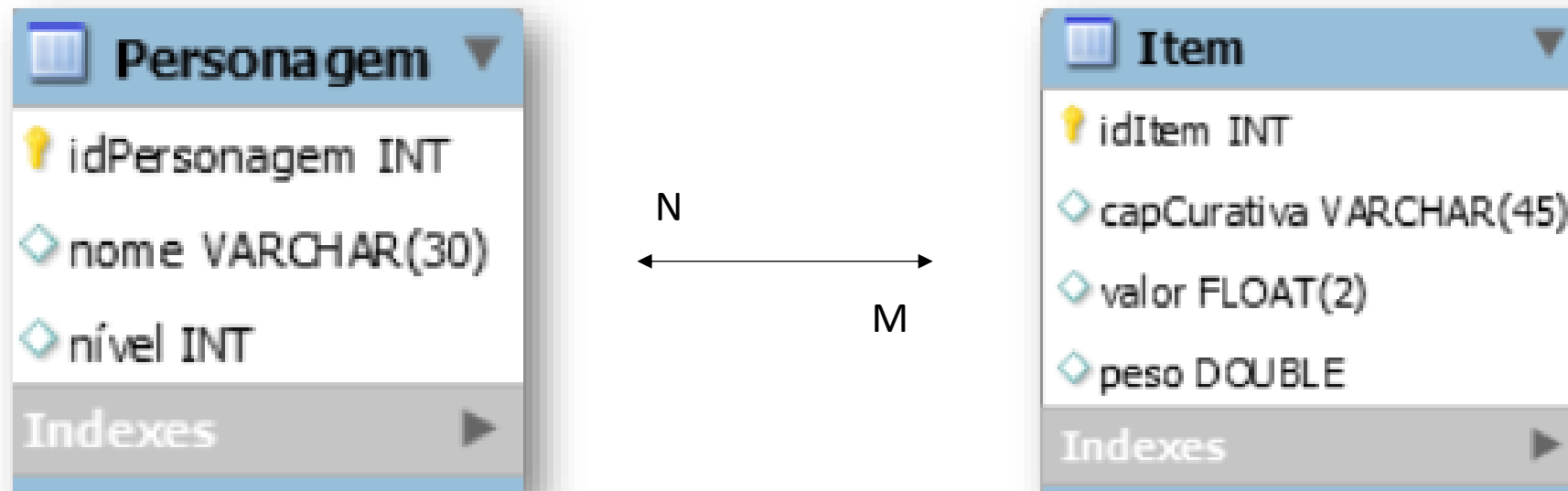


Um relacionamento 1:N adiciona uma nova coluna na tabela “N”, que armazena a chave primária da tabela “1” como uma chave estrangeira(foreign key) que é usada para fazer a relação entre tabelas

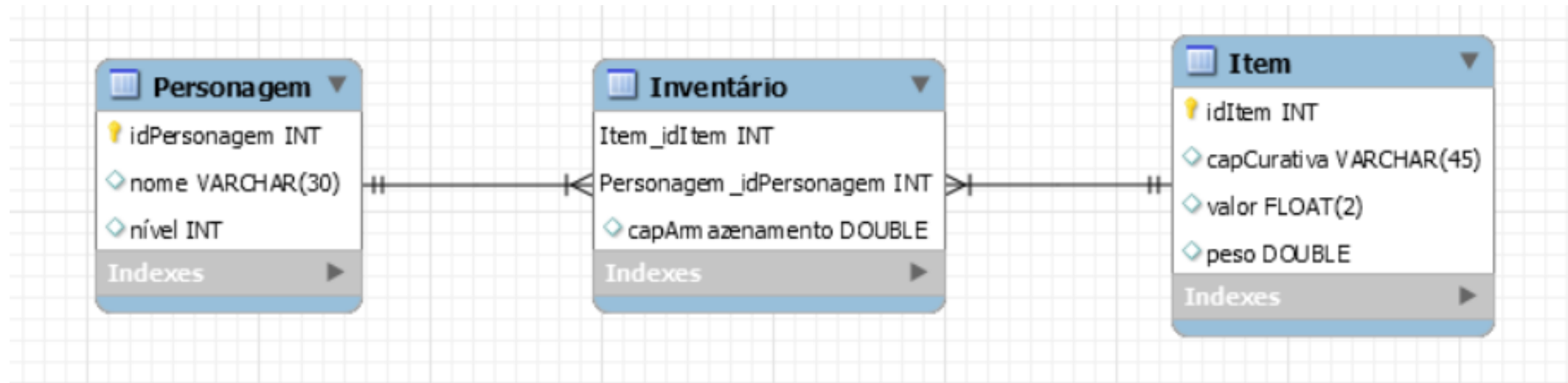
# Relacionamento N:M

Em um jogo, qual a relação entre um Personagem e um Item?

Um Personagem pode ter vários Itens(M), mas os Itens podem ser equipados por Personagens distintos(N)



# Relacionamento N:M



Nesse caso, é preciso criar uma tabela intermediária(o model cria automaticamente, mas no Script é necessária sua implementação). As chaves primárias de cada uma das entidades vira uma das chaves estrangeiras dessa tabela intermediária. É possível inserir atributos nesta tabela nova, como foi feito com “capArmazenamento”.

# 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

No reino de Hyrule, onde Link já salvou a princesa Zelda inúmeras vezes de Ganondorf, um novo empreendimento está surgindo. Inspirado pelas aventuras do herói e pela constante necessidade de itens mágicos pelos aventureiros, Link decide se aposentar temporariamente das batalhas épicas para se envolver com o comércio de itens mágicos em Hyrule. • Ele precisa modelar o banco de dados da sua futura loja de itens (mercado mágico), a HYRULE SHOP. As informações necessárias para a modelagem do banco de dados estão descritas a seguir:

- Deve-se guardar estas informações de um **item**: cada item deve ter uma identificação única, um nome, uma categoria (armas, escudos, armaduras, poções, etc.), o seu preço em rupias e a quantidade disponível no estoque.
- Deve-se guardar estas informações de uma **loja**: cada loja deve ter uma identificação única, um nome, uma localização (Vila Kakariko, Castelo de Hyrule, etc.), um responsável e a informação opcional de especialidade mágica. Um item é vendido por várias lojas diferentes e uma única loja vende um ou mais itens distintos.
- Deve-se guardar estas informações de um **cliente**: cada cliente deve ter um código de herói para identificação única, um nome, a sua região de origem (apenas texto), um cristal de comunicação mágico, uma senha única (aceita runas, números e símbolos especiais), e a informação opcional de classe de aventureiro.
- Um cliente pode comprar vários itens assim como um item é comprado por muitos clientes. Toda **compra** possui um valor total (aceita centavos de rupia), um tipo de pagamento (rupias físicas ou cristal mágico, por exemplo) e um código de rastreamento único.
- Deve-se guardar estas informações do **serviço de entrega**: o serviço de entrega deve possuir um código de identificação único, o prazo de entrega dos itens (quantidade de dias) e o valor do frete calculado (aceita centavos de rupia). Um único serviço de entrega é responsável pela entrega de todas as compras e cada compra é entregue por apenas esse serviço.
- Após a entrega, um cliente pode avaliar diversos itens e cada item pode ser avaliado por vários clientes. Toda **avaliação** feita pelo cliente deve possuir uma nota de um a cinco corações. O cliente tem a opção de escrever um comentário para esclarecer a nota destinada ao item.



AULA  
CONCLUÍDA!