

CURSO

Administrador de Banco de Dados

Disciplina II

Modelagem de Banco de Dados



AULA 09

Modelo Lógico

- Tipos de Dados
- Chaves Estrangeiras



Introdução

- O **modelo lógico** busca demonstrar as **entidades** e **relacionamentos** do banco de dados através de uma nova perspectiva, **incluindo** duas informações importantes:
 - Os tipos dos dados salvos na tabela;
 - As chaves estrangeiras que unem as tabelas relacionadas;

Tipo de Dados

- O **tipo de dado** salvo em uma tabela diz respeito ao **formato** do dado em cada **atributo** da entidade;
- Cada **atributo** possui um **tipo de dado** específico;
- Exemplos comuns:
 - *Data de Nascimento*: **Data**
 - *Nome*: **Texto**
 - *Salario*: **Número Fracionado**
 - *Idade*: **Número Inteiro**

Tipo de Dados

- No **modelo lógico** o **tipo de dado** dos atributos é identificado;
- O **nome do tipo** de dado irá **variar** de acordo com o **SGBD** escolhido para o sistema de banco de dados;
- No MySQL os **tipos de dados** mais comuns são:
 - **Texto: `varchar (100)`**
 - **Data: `date`**
 - **Hora: `time`**
 - **Número Fracionado: `double` ou `float`**
 - **Número Inteiro: `integer`**

Varchar

- O tipo **varchar** (texto) recebe **qualquer tipo de caractere**, seja letras, números ou símbolos.
- Porém, para o SGBD o **tipo do dado** sempre será texto;
- **Por exemplo:**
 - ✓ Você tem um atributo chamado salario e pretende fazer cálculos matemáticos com ele, o tipo **não deve** ser varchar, pois não podemos fazer cálculos com texto mas sim com números;
 - ✓ Já se você tem um atributo numérico mais quer armazenar símbolos junto com o número você deve escolher varchar;

Varchar

- O tipo varchar vem acompanhado com um número dentro de parênteses. **Exemplo:** varchar (100);
- Esses número dentro do parêntese é o **tamanho** do texto, ou seja, **quantos caracteres** este texto poderá possuir;
- **Exemplo:** **nome varchar (10)** -> significado que o nome dentro deste atributo da tabela não pode ter mais do que 10 caracteres;
- O tipo varchar precisa de **aspas simples** entre o texto inserido no atributo. **Exemplo:** *'Jackson Henrique'*;

Date

- O tipo date é utilizado somente para **datas** e não aceita receber números, textos ou horas;
- **Exemplo:** **data_nascimento date;**
- No MySQL as datas são inseridas no banco de dados de uma forma diferente: **de trás para frente**;
- O tipo date também **precisa de aspas simples** entre o valor da data, senão o MySQL não entende o sinal de – (traço);
- **Exemplo:** *'2020-08-31'*

Time

- O tipo time é utilizado **somente para horas** e não aceita receber números, textos ou datas;
- **Exemplo: `hora_inicio time;`**
- No MySQL as datas são inseridas no banco de dados de uma forma **específica usando a sequencia, hora, minuto e segundo;**
- Como o tipo date o tipo time **também precisa de aspas simples** entre os valor inserido;
- **Exemplo: `'13:56:00'`**

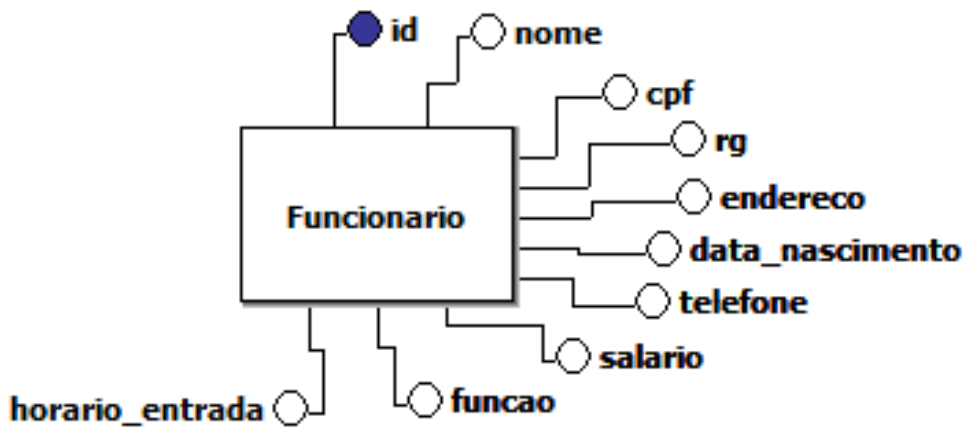
Float ou Double

- Os tipos Float e Double são usados para **números fracionados**, ou seja número que precisam demonstrar um valor com **casas decimais**;
- Normalmente é utilizado para **valores financeiros** para demonstrar os centavos por exemplo. **Exemplo: salario float;**
- **Exemplo do número fracionado: 500,50**
- No MySQL este formato **não precisa de aspas simples**, porém ao inserir um número **não temos a vírgula ao separar a casa decimal mais sim um ponto**;
- **Exemplo do número fracionado no MySQL: 500.50**
- Para separa a casa do milhar, **não temos sinal**.
- **Exemplo: 1500.50**

Integer

- O tipo integer serve para armazenar um número inteiro, que **não possui casa decimal**;
- São normalmente usados para demonstrar um **dia, mês ou ano**. Ou um número **simples**;
- **Exemplo: 1990 ou 06 ou 50**
- No MySQL este formato **não precisa de aspas simples**;
- Os **atributos identificadores** sempre são do tipo **inteiro**;

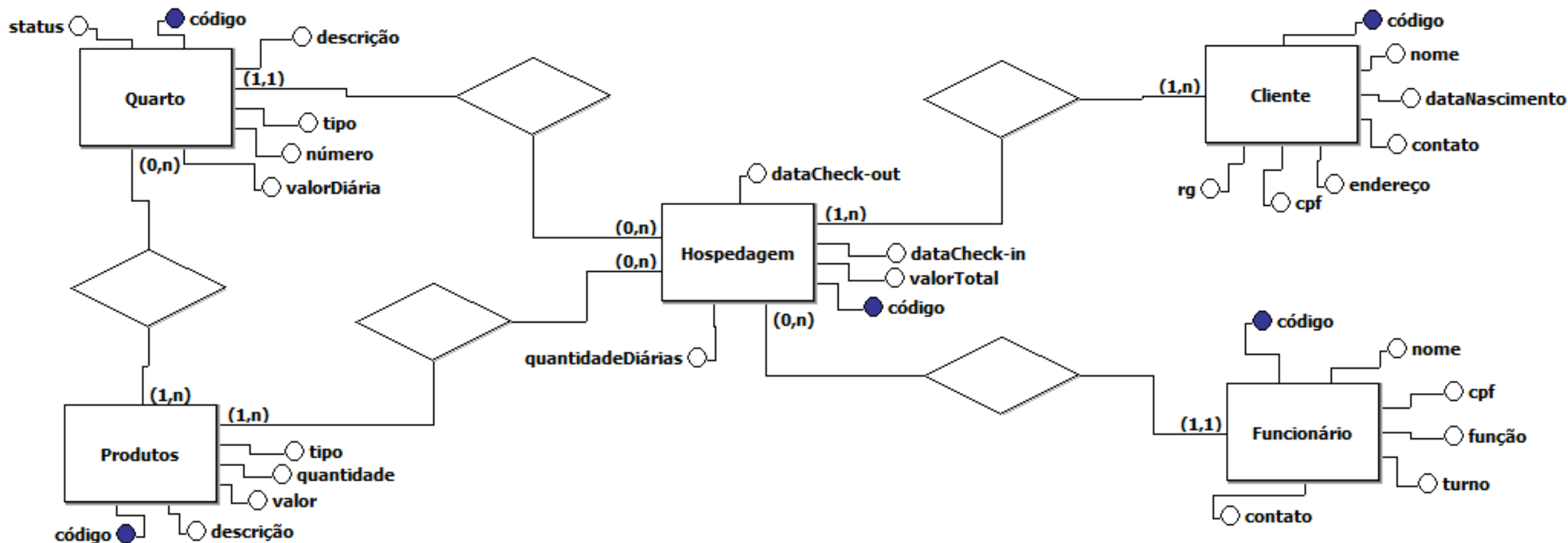
Exemplo Prático



Atributos	Tipo	Exemplo:
id	integer	1
nome	Varchar (200)	'Jackson'
cpf	Varchar (30)	'123.123.123.15'
rg	Varchar (50)	'881231 sesdec/RO'
endereco	Varchar (250)	'Av. X, 111 – Vida'
Data_nascimento	Date	'1987-06-30'
telefone	Varchar (100)	'69 98408 5712'
salario	Float	30500.50
funcao	Varchar (40)	'Professor'
horario_entrada	Time	'08:30:00'

Hora de praticar

Identifique os tipos de dados

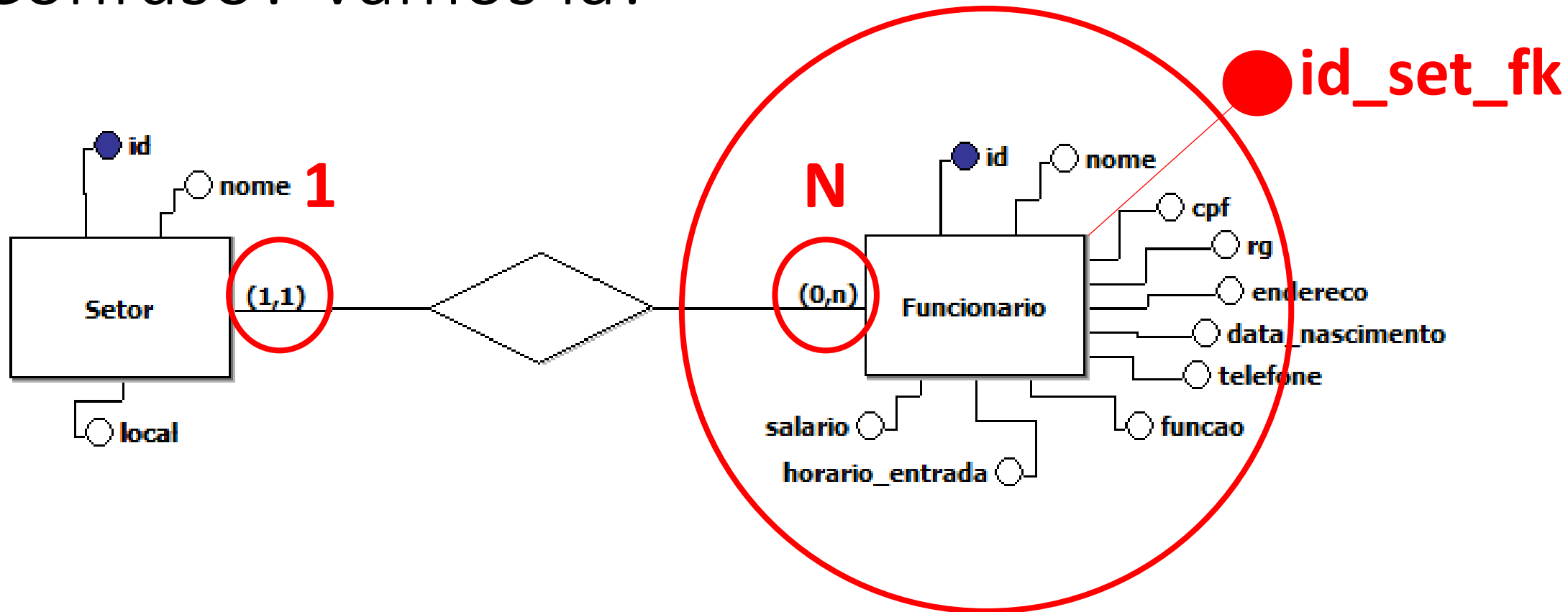


Chaves Estrangeiras

- Até aqui conhecemos **dois tipos de atributos** no banco de dados: o **atributo simples** e o **identificador** também chamado de *Primary Key* (PK) ;
- Agora iremos conhecer um **terceiro tipo**, a **chave estrangeira**;
- A **chave estrangeira** ou *Foreign Key* (FK) é um atributo criado na **tabela que recebe o N** entre duas tabelas relacionadas;
- Este atributo tem como objetivo **arramar a relação entre duas tabelas**;
- A **chave estrangeira** (FK) corresponde ao **atributo identificador** (PK) da **tabela com a cardinalidade 1** do relacionamento;
- As **FK** sempre são do **tipo Inteiro**, uma vez que são representações das **PKs**;

Confuso? Vamos lá!

A tabela com o N recebe a PK da tabela com 1



- A FK é a PK da tabela com o 1
- set é abreviação de Setor

Atributos	Tipo	Exemplo:
id	integer	1
nome	Varchar (200)	'Jackson'
cpf	Varchar (30)	'123.123.123.15'

Atributos	Tipo	Exemplo:
id	integer	1
nome	Varchar (200)	'Jackson'
cpf	Varchar (30)	'123.123.123.15'
rg	Varchar (50)	'881231 sesdec/RO'
endereço	Varchar (250)	'Av. X, 111 – Vida'
Data_nascimento	Date	'1987-06-30'
telefone	Varchar (100)	'69 98408 5712'
salario	Float	30500.50
funcao	Varchar (40)	'Professor'
horario_entrada	Time	'08:30:00'
Id_set_fk	Integer	1

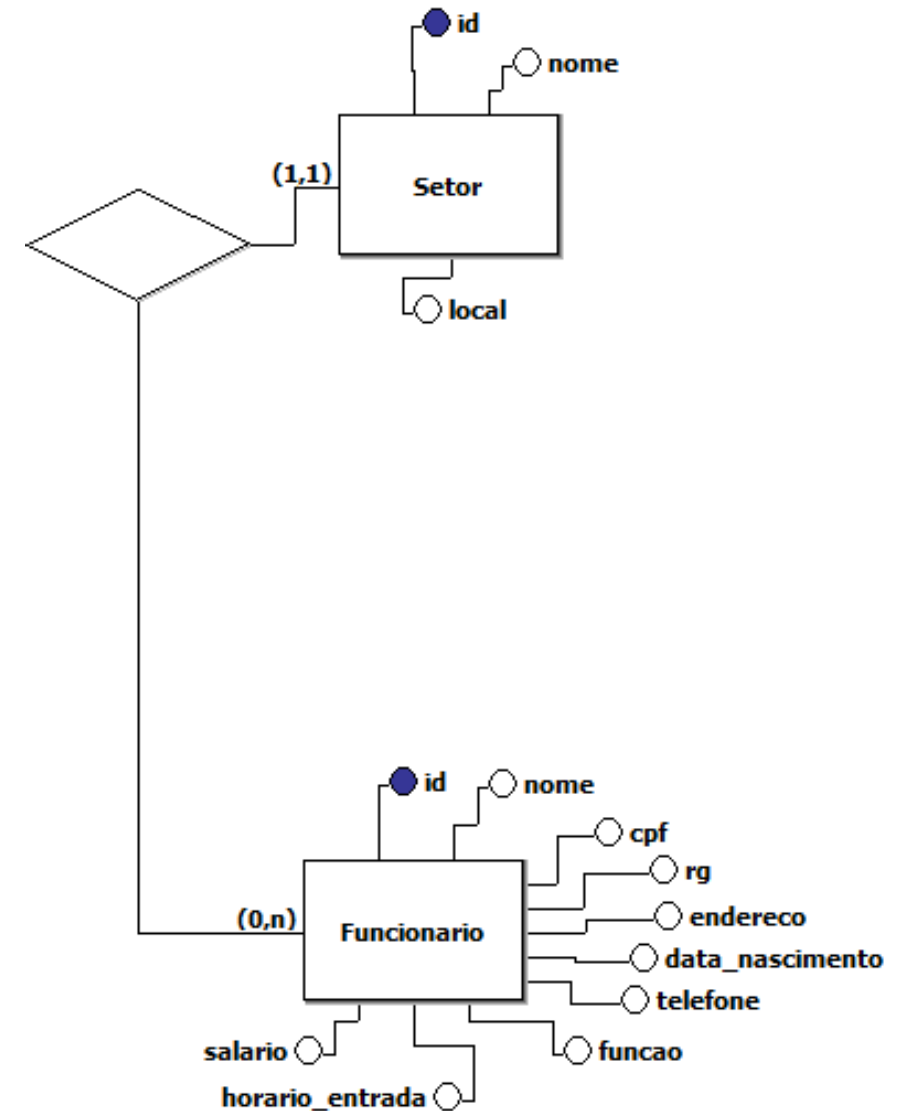


Tabela: Setor

Id	Nome	Local
1	'Vendas'	'Frente'
2	'Financeiro'	'Fundo'
3	'Almoxarifado'	'Fundo'

Tabela: Funcionário

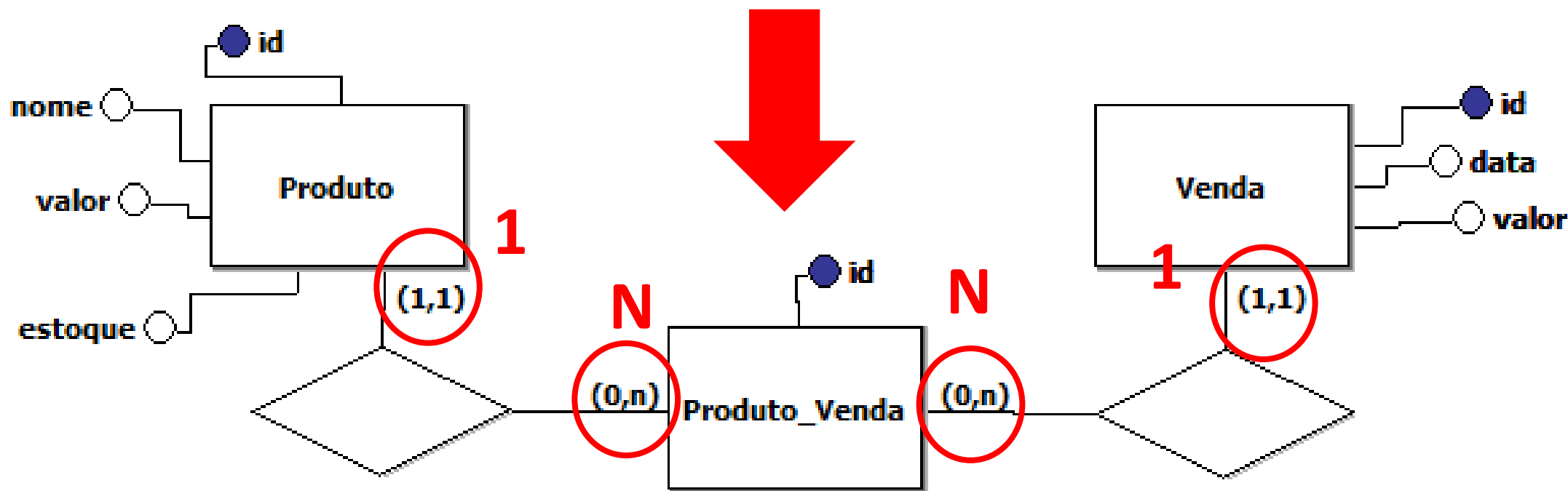
Id	Nome	Cpf	Rg	Endereco	Data_nasc	Telefone	Funcao	Horario_entrada	salario	Id_set_fk
1	'Jackson Henrique da Silva bezerra'	'123.123.522-22'	'445588 sesdec-RO'	'Rua X, 522 – são Paulo'	'1987-06-30'	'69 98408-5712'	'Vendedor'	'08:00:00'	10500.50	1
2	'Ana Paula Silva'	'123-123-123-23'	'123123 sesdec-RO'	'Rua A, 223 – são Pedro'	'1990-01-01'	'69 98408-5555'	'Caixa'	'09:00:00'	5000.00	2

Lembre-se

- A **chave estrangeira** só existe a partir do **modelo lógico**;
- Mas e as tabelas **N para N** como fica?
- E as **1 para 1**?

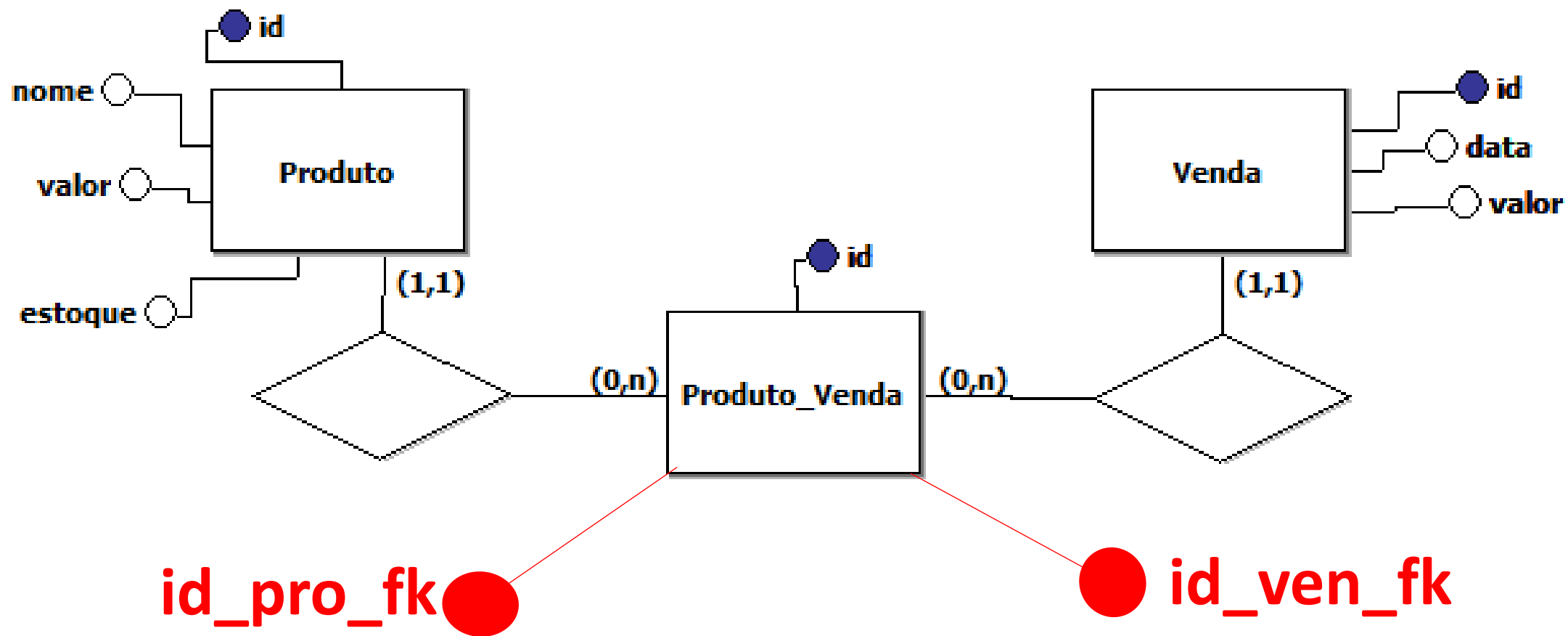
Desmembramento de tabela N para N

- Lembre-se: o relacionamento **N para N** só existe no modelo conceitual;
- A partir do **modelo lógico** não temos relacionamento N para N, apenas **N para 1**;
- Para resolver isso temos que **desmembrar** o relacionamento **N para N** para que o mesmo fique **N para 1**;
- Isso é feito **criando uma 3ª tabela** no meio do relacionamento entre as duas tabelas **N para N**;
- Como?



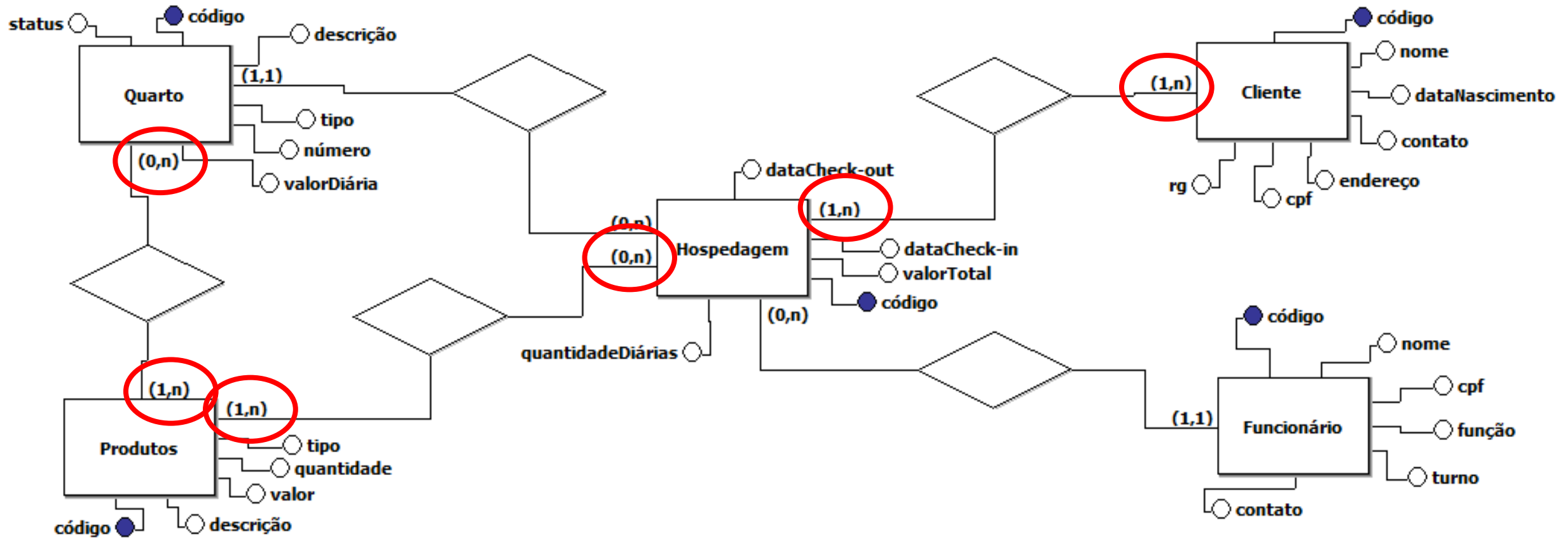
Regras do N para N

1. Quando houver N para N **cria-se** a 3ª Tabela, ligada as duas tabelas de origem;
2. A **3ª Tabela** recebe o **nome** das duas tabelas de origem;
3. A **3ª Tabela** recebe o **N** na relação com as duas tabelas de origem;
4. Já as **tabelas de origem** ficam com o **1**;
5. Assim a **3ª Tabela** recebe as **chaves estrangeiras** das tabelas de origem;



Hora de Praticar

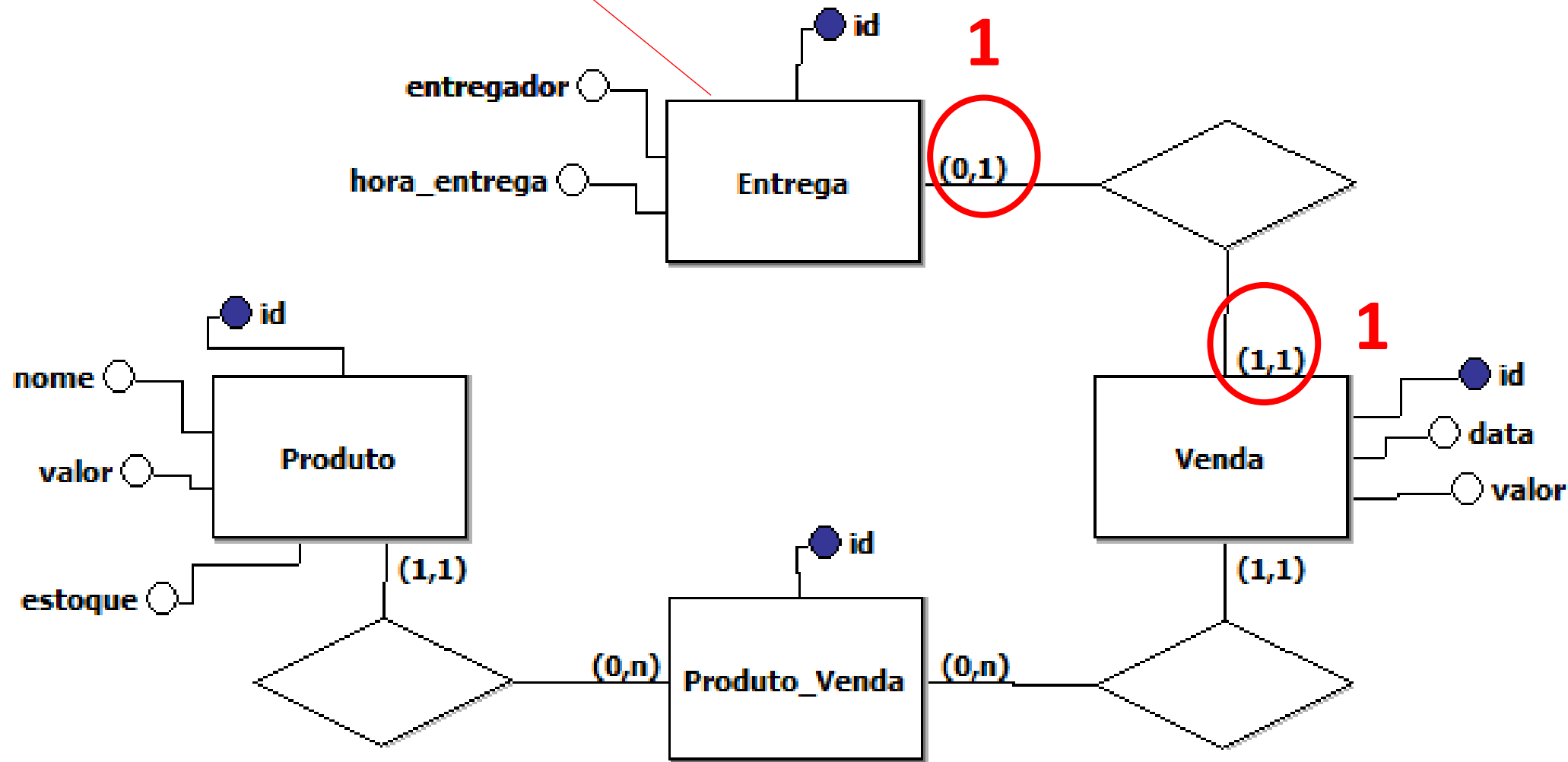
- Desmembre o N para N deste diagrama;



E o 1 para 1 com fica?

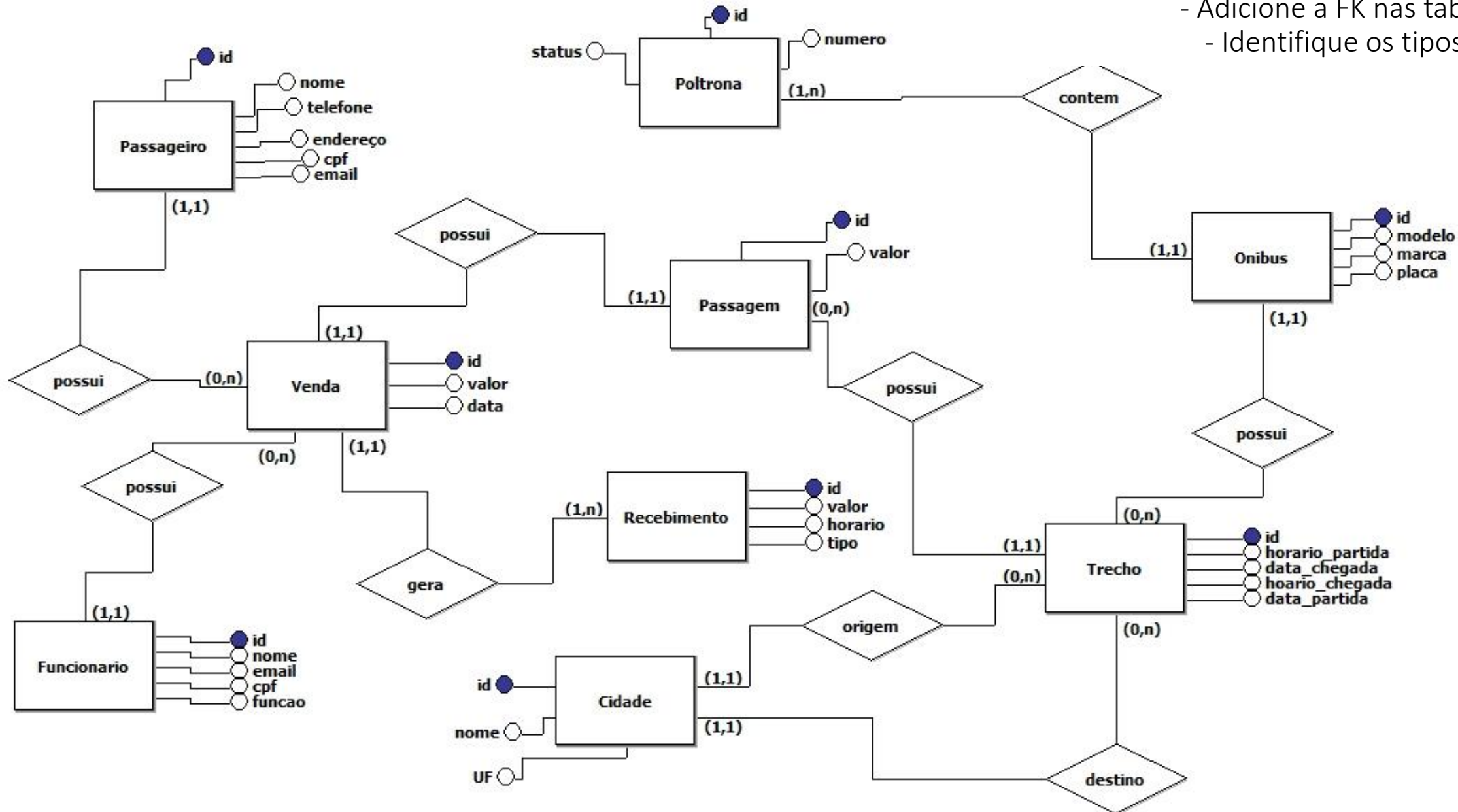
- No relacionamento **1 para 1** você deve **escolher uma das duas tabelas** para receber a **chave estrangeira**;
- Na prática o **único relacionamento** existente no **modelo físico e lógico** é o **1 para N**;
- No **1 para 1**, escolha a tabela que faça **mais sentido** receber a chave da outra;

id_ven_fk ●



Hora de Praticar 2

- Desmembre o N para N deste diagrama;
- Adicione a FK nas tabelas N para 1
- Adicione a FK nas tabelas 1 para 1
- Identifique os tipos dos dados



Jackson Henrique

Professor Formador

E-mail:

Jackson.henrique@ifro.edu.br

