

Objetivos:

- I. Introdução ao modelo lógico;
Modelo relacional;
- II. Chave estrangeira;
- III. Restrições de integridade;
- IV. Conversão do DER para o modelo relacional.

I. Introdução ao modelo lógico

O modelo lógico de dados leva os elementos da modelagem conceitual de dados um passo adiante, adicionando mais informações a eles.

Um modelo de dados lógico estabelece a estrutura dos elementos de dados, os relacionamentos entre eles e algumas restrições.

Um modelo de dados lógico deve ser projetado para ser independente da tecnologia, de modo a não ser afetado pelas rápidas mudanças do ambiente de negócios. O modelo lógico geralmente adota o modelo relacional (baseada no modelo Entidade Relacionamento) ou modelo orientado a objetos (baseada na UML – Linguagem Unificada de Modelagem).

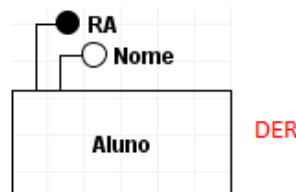
Minimundo a ser modelado



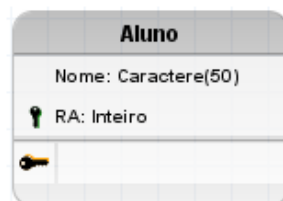
RA: 123
Nome: Pedro

Modelo conceitual

Aluno: RA e Nome
MER



Modelo lógico



Representação de
uma tabela

Modelo físico

```
create table tbaluno(  
    ra integer not null primary key,  
    nome varchar(50)  
);
```

Cláusula SQL compatível com o
SGBD PostgreSQL

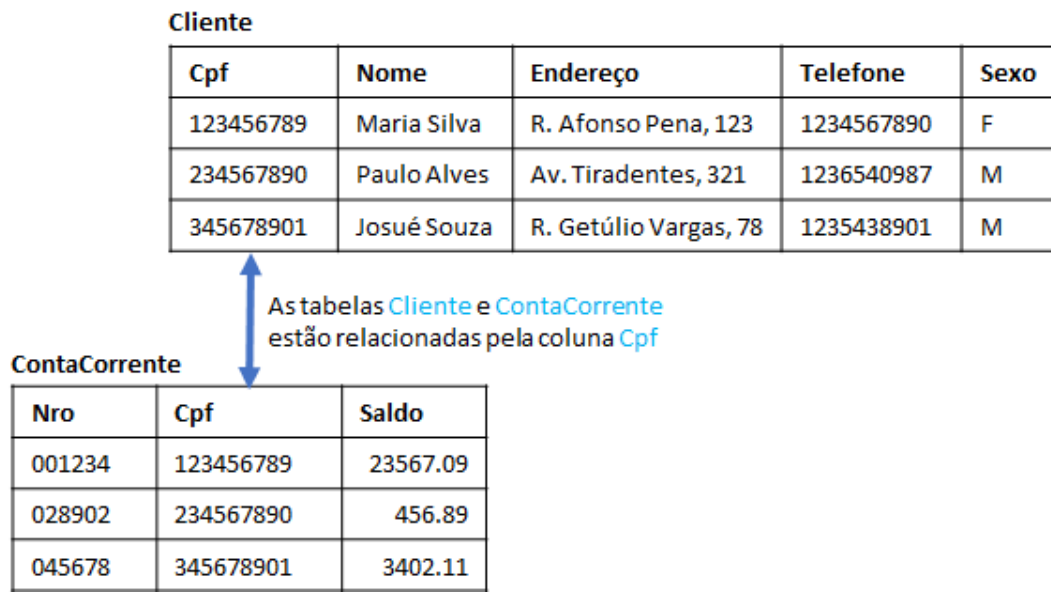
II. Modelo relacional

O modelo relacional foi criado por Codd em 1970 e tem por finalidade representar os dados como uma coleção de relações, onde cada relação é representada por uma tabela, ou falando de uma forma mais direta, um arquivo.

O modelo relacional foi o primeiro modelo de dados para SGBD comerciais.

É usado na maioria das aplicações comerciais. Os SGBD relacionais são baseados no modelo relacional.

O BD relacional é constituído por uma coleção de tabelas, compostas por colunas e linhas, inter-relacionadas:



Na terminologia do modelo relacional:

- **tabela** é chamada de **relação**;
- **Linha** ou **registro** de uma tabela é chamada de **tupla**;
- **Coluna** ou **campo** é chamado de **atributo**;
- O tipo de dado que especifica o tipo dos valores que podem aparecer em uma coluna é chamado de **domínio**.

O domínio de um atributo é o conjunto de possíveis valores que o atributo pode receber. No exemplo a seguir é exibido os domínios dos atributos da tabela Cliente:

Atributo	Domínio
Cpf	Inteiro longo sem sinal
Nome	Conjunto de caracteres
Endereço	Conjunto de caracteres
Telefone	Conjunto de caracteres
Sexo	Conjunto de caracteres

O domínio dos atributos de uma tabela (relação) é semelhante aos tipos de dados das variáveis na programação.

Diferença entre esquema e instância: a descrição da tabela é chamada de **esquema** da tabela os dados armazenados na tabela formam um conjunto chamado de **instâncias** da tabela. A seguir tem-se destacado o **esquema** e as **instâncias** da relação (tabela) Cliente:

Esquema da relação Cliente

Cpf	Nome	Endereço	Telefone	Sexo
123456789	Maria Silva	NULL	1234567890	F
234567890	Paulo Alves	Av. Tiradentes, 321	NULL	M
345678901	Josué Souza	R. Getúlio Vargas, 78	1235438901	M

Instâncias da relação Cliente

III. Chave estrangeira

A chave estrangeira é uma coluna ou combinação de colunas, cujos valores aparecem necessariamente na chave primária de uma outra tabela.

Este é o mecanismo que permite a implementação de relacionamentos no BD relacional. O fato da coluna **Cpf** da tabela **ContaCorrente** utilizar os mesmos valores armazenados na coluna **Cpf** da tabela **Cliente** forma o conceito de relacionamento (ligação) entre as tabelas. Através do relacionamento, evitamos a repetição de dados.

Cliente				
Cpf	Nome	Endereço	Telefone	Sexo
123456789	Maria Silva	R. Afonso Pena, 123	1234567890	F
234567890	Paulo Alves	Av. Tiradentes, 321	1236540987	M
345678901	Josué Souza	R. Getúlio Vargas, 78	1235438901	M

As tabelas **Cliente** e **ContaCorrente** estão relacionadas pela coluna **Cpf**

ContaCorrente		
Nro	Cpf	Saldo
001234	123456789	23567.09
028902	234567890	456.89
045678	345678901	3402.11

Nesse exemplo **Cpf** é a chave primária da tabela **Cliente** e **Nro** é chave primária da tabela **ContaCorrente**.

Na tabela **ContaCorrente** **Cpf** é chave estrangeira e só poderá receber valores que existam na coluna **Cpf** da tabela **Cliente**.

Observações sobre uma chave estrangeira:

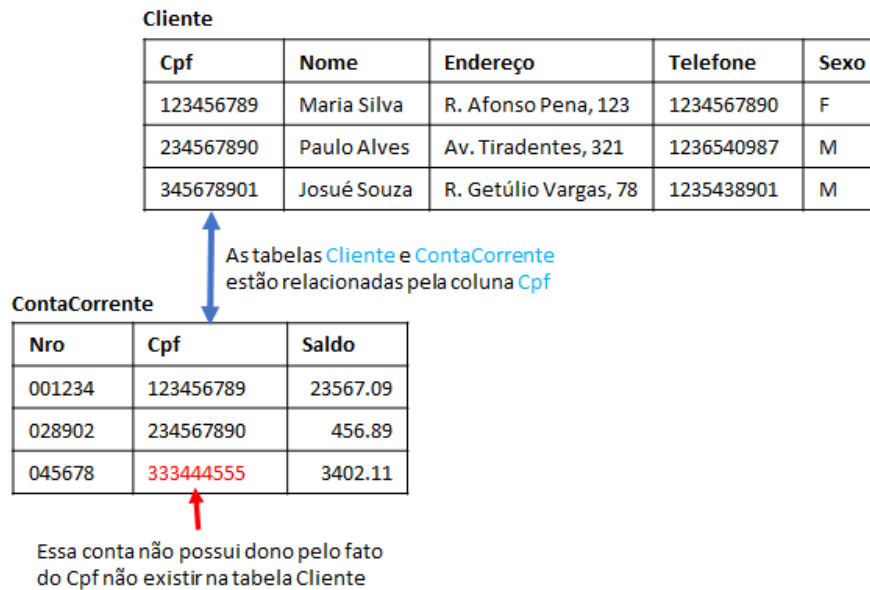
- Uma chave estrangeira não precisa ser uma chave primária na sua relação. Por exemplo, o atributo Cpf compõe a chave primária da tabela ContaCorrente;
- Uma chave estrangeira não precisa ter o mesmo nome que a chave primária correspondente na outra tabela (apenas o mesmo domínio). Por exemplo, o atributo chave estrangeira Cpf na tabela ContaCorrente poderia ser nomeado com outro nome, tal como, CódigoCliente.

Restrições que devem ser garantidas ao serem executadas as operações de inserir, alterar e remover na chave estrangeira:

- Inserir registro/tupla na tabela que contém a chave estrangeira:
 - É necessário garantir que o valor exista na chave primária da tabela referenciada.
- Alterar o valor da chave estrangeira na tabela referenciada:
 - É necessário garantir que o valor seja atualizado na chave estrangeira.
- Remover o valor da chave estrangeira na tabela referenciada:
 - É necessário garantir que o valor não exista numa chave estrangeira.

IV. Restrições de integridade

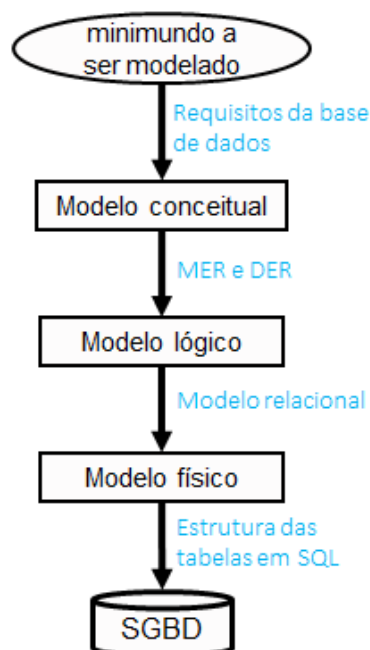
- **Restrição de domínio** determina que o valor de cada atributo de um esquema de relação deve ser um valor do domínio do atributo. Por exemplo, a coluna Cpf deve receber somente valores do domínio “inteiro longo sem sinal”;
- **Restrição de chave** determina que os valores dos atributos que compõem uma chave primária devem ser únicos de modo a identificar o registro. Por exemplo, a coluna Cpf da tabela Cliente não poderá ter valores repeditos;
- **Restrição de integridade de entidade** determina que nenhum valor da chave primária pode ser nulo. Isso porque, o valor de uma chave primária é utilizado para identificar os registros da relação. Por exemplo, se duas ou mais tuplas tiverem o valor null na chave primária, não haverá como diferenciar um registro do outro;
- **Restrição de integridade referencial** é uma restrição usada para manter a consistência entre os registros de duas relações. Ela estabelece que um registro de uma relação que se refere à outra relação deve-se referir a um registro existente naquela relação. No exemplo a seguir, na coluna Cpf da tabela ContaCorrente não pode existir um registro cujo valor do Cpf não exista na tabela Cliente.



V. Conversão do DER para o modelo relacional

Em projetos de BD é comum realizarem a modelagem de dados através de um modelo de alto-nível. Os produtos gerados por esse processo são os esquemas de visões que são posteriormente integrados para formar um único esquema.

Passos de um projeto de BD



O modelo de dados de alto-nível normalmente adotado é o MER e o esquema das visões e de toda a base de dados são especificados em DER.

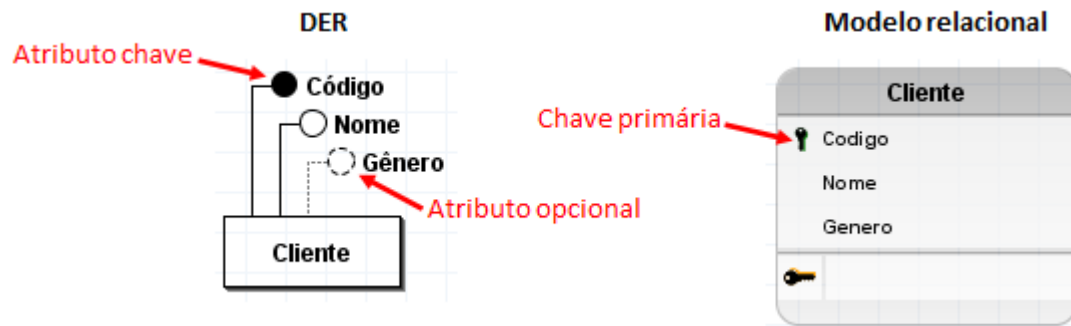
O passo seguinte à modelagem dos dados é o mapeamento do DER para um modelo de dados de implementação. Existem os modelos de dados de implementação: hierárquico, rede, relacional e orientado a objetos. Para cada um desses modelos, pode-se definir estratégias de tradução a partir de um DER. Aqui faremos a tradução para o modelo relacional.

Tendo em mente as fases de construção de um banco de dados, o MER e DER são utilizados durante o projeto conceitual e o modelo de dados relacional durante o projeto lógico.

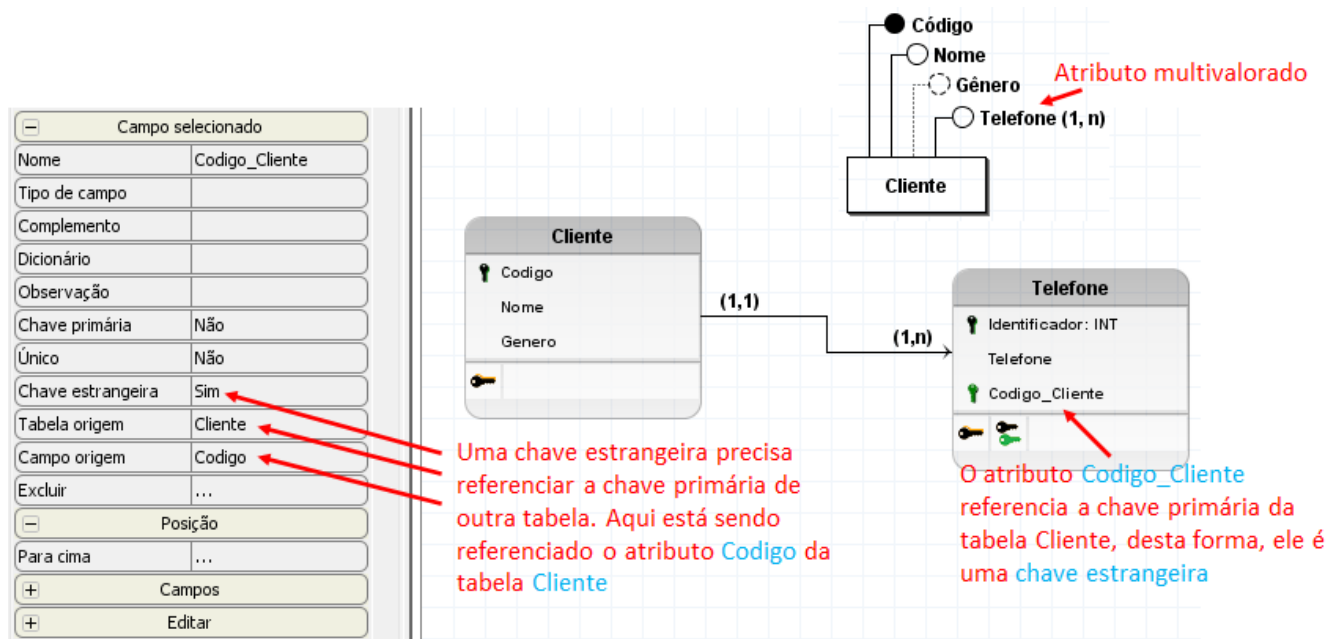
Conversão de entidade: o tipo de entidade é traduzido para uma relação (tabela). No exemplo a seguir o tipo de entidade **Cliente** foi traduzido para a representação de esquema de tabela no modelo relacional. Observações:

- O atributo chave é chamado de chave primária no modelo relacional. O atributo **Código** é a chave primária da tabela **Cliente**;

- O atributo opcional é representado normalmente no modelo relacional.



Conversão de atributo multivalorado: o atributo multivalorado se torna uma nova relação (tabela) contendo a chave estrangeira que faz ligação com a tabela que deveria possuir o atributo multivalorado. No exemplo a seguir o atributo multivalorado Telefone se tornou a tabela Telefone. A ligação entre as tabelas se dá pela chave estrangeira `Codigo_Cliente` que faz a ligação com a chave primária da tabela Cliente.



Como cada telefone será 1 registro e 1 cliente pode ter N telefones, então não é possível acomodar N telefones em 1 registro de cliente. Por este motivo é necessário colocar os registros de telefones numa tabela específica.

A figura a seguir é usada para ilustrar como os dados dos clientes serão armazenados no BD. Veja que cada telefone ocupa um único registro (linha da tabela).

Tabela Cliente

Codigo	Nome	Genero
1	Ana	F
2	Pedro	M
3	Maria	F

Tabela Telefone

Identificador	Telefone	Codigo_Cliente
1	1239012345	1
2	12987654321	3
3	1239005522	3
4	12912122323	3

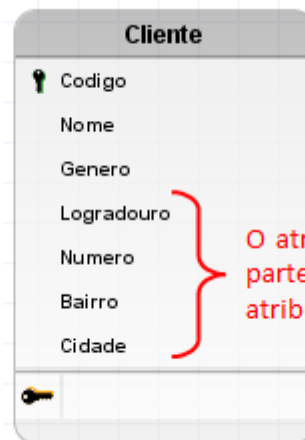
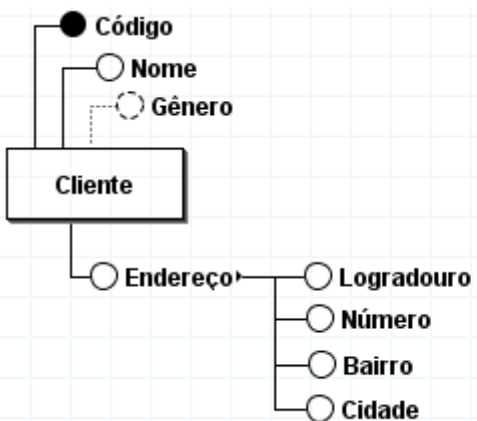
Estes três telefones pertencem a Maria

A ligação entre as tabelas se dá pela chave estrangeira Codigo_Cliente

No relacionamento entre Cliente e Telefone a chave estrangeira migra sempre para o lado N.

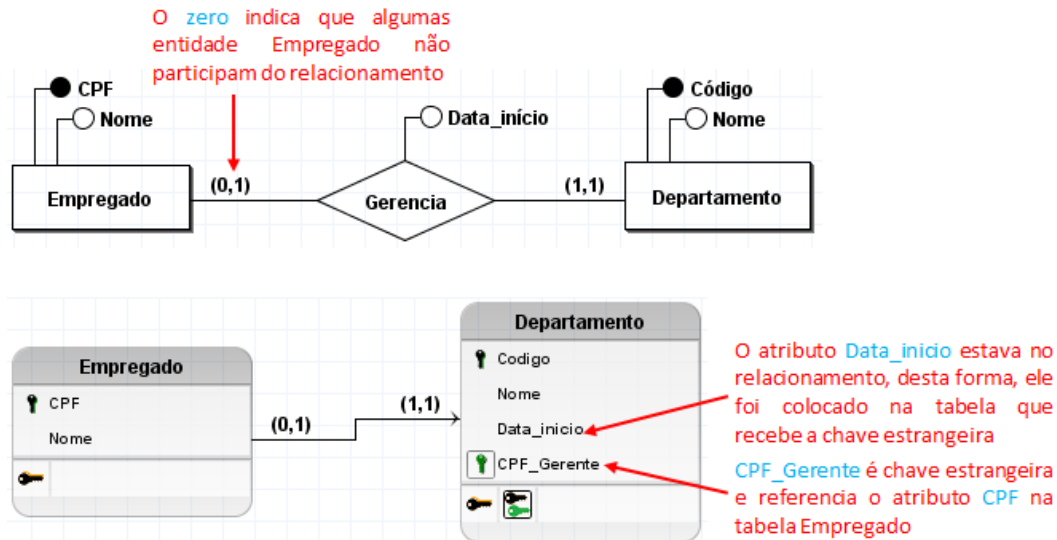
A chave estrangeira é o único instrumento de ligação entre duas tabelas.

Conversão de atributo composto: no modelo relacional são colocados apenas os atributos simples. No exemplo a seguir o atributo **Endereço** não foi incluído no modelo relacional, mas os atributos simples que compõem o atributo composto foram incluídos na tabela.



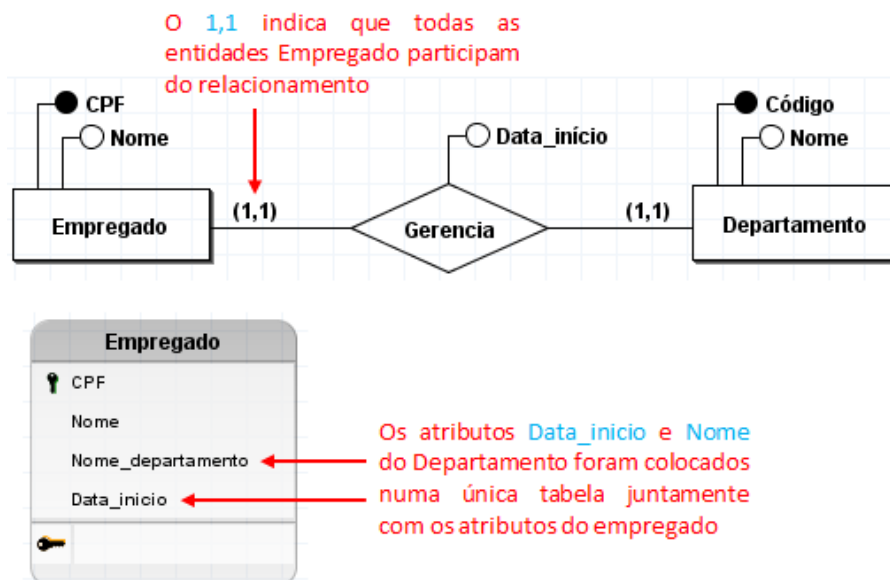
Conversão de relacionamento 1:1: para cada tipo de relacionamento 1:1 do DER:

- Criar as relações **S** e **T** que correspondem aos tipos de entidades participantes de **R**. No exemplo a seguir foram criadas as relações **Empregado** e **Departamento**;
- Escolher uma das relações - é melhor escolher o tipo de entidade com participação total em **R**. Digamos **T**, e incluir como **chave estrangeira** de **T** a **chave primária** de **S**. Incluir todos os atributos simples de **R** como atributos de **T**. No exemplo a seguir foi incluída a **chave estrangeira** **CPF_Gerente** que referencia a chave primária da tabela **Empregado**.



Os atributos que estão no relacionamento precisam migrar para uma das tabelas. Como o relacionamento é 1:1 entre as entidades **Empregado** e **Departamento**, então o atributo **Data_inicio** e a chave estrangeira poderiam ser colocados em qualquer uma das tabelas. Porém, todo departamento deve ter um gerente, mas nem todos os empregados são gerentes, então considera-se que a participação de **Departamento** no relacionamento é **total**, por este motivo o atributo **Data_inicio** e a chave estrangeira foram colocado na tabela **Departamento**.

No exemplo a seguir considera-se que todo empregado gerencia um departamento, logo a participação de ambas as entidades é total. A solução neste caso é criar uma única tabela com todos os atributos.



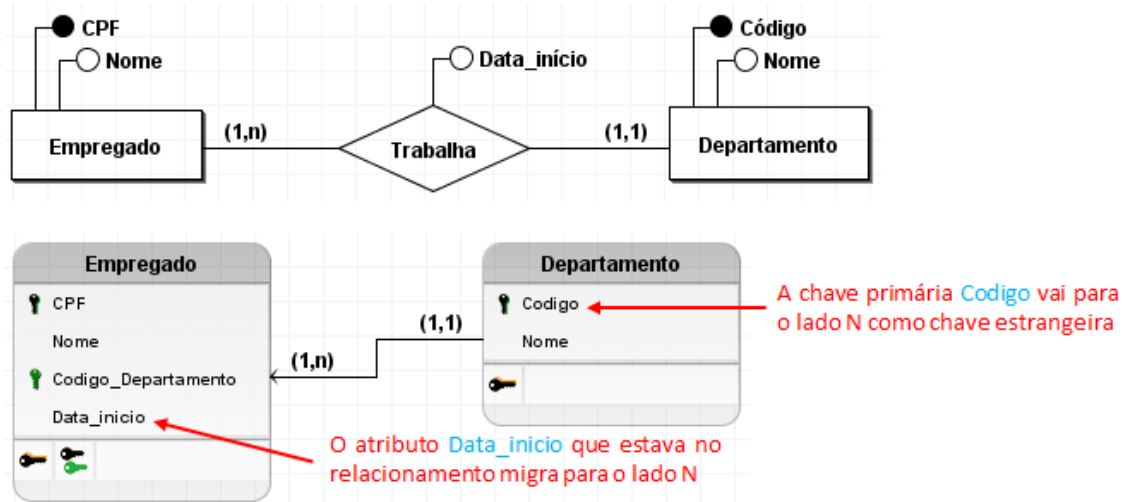
Conversão de relacionamento 1:N: para cada tipo de relacionamento **R** 1:N do DER:

- Identificar a relação **S** que representa o tipo de entidade que participa do lado **N** do tipo de relacionamento;
- Incluir como chave estrangeira de **S** a chave primária da relação **T**, que representa o outro tipo de entidade que participa em **R**, isto porque cada instância de entidade do lado **1** está relacionada a mais de

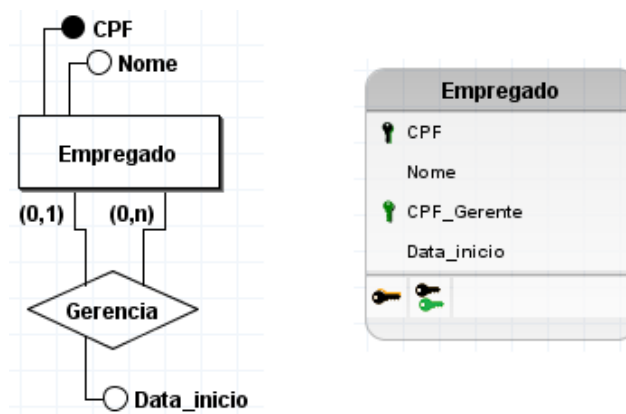
uma instância de entidade no lado **N**. Por exemplo, no relacionamento **1:N** Trabalha cada empregado está relacionado a um único departamento;

- Incluir também quaisquer atributos do tipo de relacionamento **1:N** como atributos de **S**.

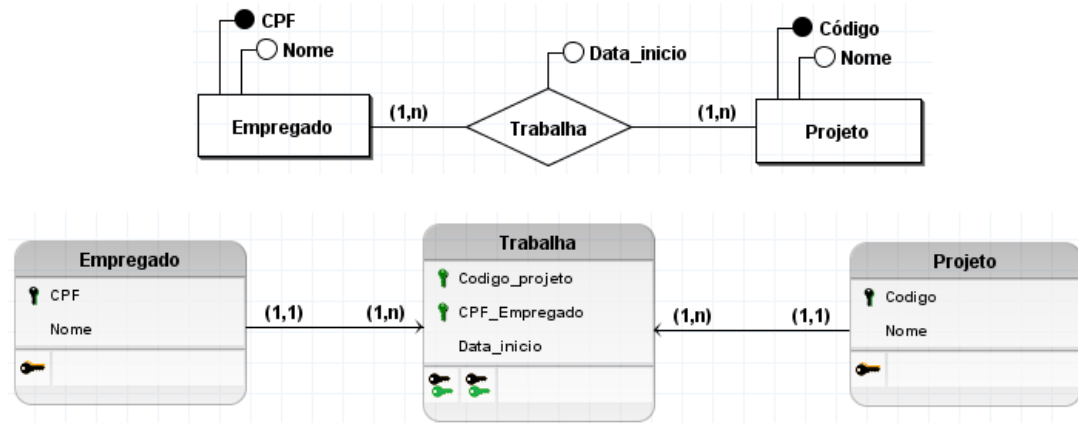
No exemplo a seguir 1 empregado trabalha para 1 departamento, e 1 departamento pode ter N empregados. Desta forma, a chave primária do lado 1 vai para o lado N como chave estrangeira, bem como o atributo Data_inicio que está no relacionamento Trabalha.



Conversão de relacionamento 1:N recursivo: no exemplo a seguir, alguns empregados são gerentes, isso está representado pela cardinalidade **0,1**, e um gerente pode ter N subordinados, representado pela cardinalidade **0,n**. Como o relacionamento é 1:N é recursivo, então a chave primária CPF migra para a tabela Empregado como chave estrangeira, porém veja que **CPF** e **CPF_gerente** são atributos distintos. O atributo Data_inicio também migra para o lado N.



Conversão de relacionamento N:N: o relacionamento **N:N** se torna uma tabela. No exemplo a seguir, o relacionamento **Trabalha** se tornou uma tabela que recebe as chaves estrangeiras das tabelas Empregado e Projeto, bem como o atributo Data_inicio do relacionamento.



Exercícios

Exercício 1: Qual é a diferença entre esquema e instância no modelo relacional?

Exercício 2: Qual é o mecanismo utilizado para implementar o relacionamento no modelo relacional?

Exercício 3: Converter do modelo conceitual para o modelo lógico o Exercício 3 da Aula 2.

Exercício 4: Converter do modelo conceitual para o modelo lógico o Exercício 4 da Aula 2.

Exercício 5: Converter do modelo conceitual para o modelo lógico o Exercício 5 da Aula 2.

Exercício 6: Converter do modelo conceitual para o modelo lógico o Exercício 6 da Aula 2.

Exercício 7: Converter do modelo conceitual para o modelo lógico o Exercício 7 da Aula 2.

Exercícios Complementares

Exercício 8: Converter do modelo conceitual para o modelo lógico o Exercício 8 da Aula 2.

Exercício 9: Converter do modelo conceitual para o modelo lógico o Exercício 9 da Aula 2.

Exercício 10: Converter do modelo conceitual para o modelo lógico o Exercício 10 da Aula 2.