

Modelo Logico

Parte I

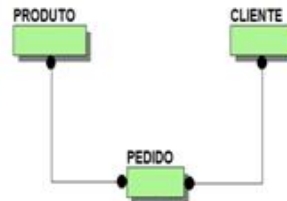
MODELO LÓGICO

1. Conceito;
 2. Tabelas;
 3. Chave Primaria (PK – Primary Key);
 4. Chave Única (Unique);
 5. Chave Estrangeira (FK – Foreign Key);
 6. Relacionamentos;
 7. Tipos de Relacionamentos
 8. Notação Resumida
 9. Tipos de Integridades;
- Integridade de Dados;**
Integridade de Domínio;
Integridade de Entidade;
Integridade Referencial.

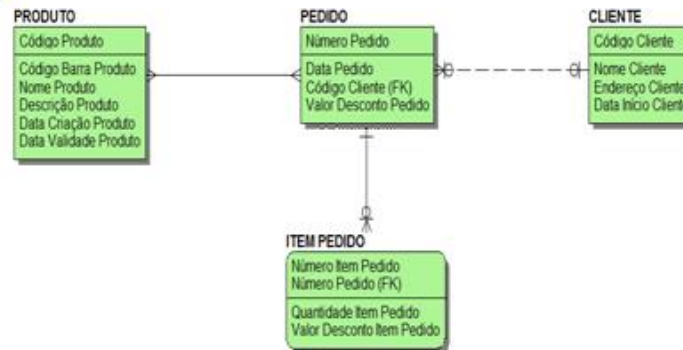


NEGÓCIO

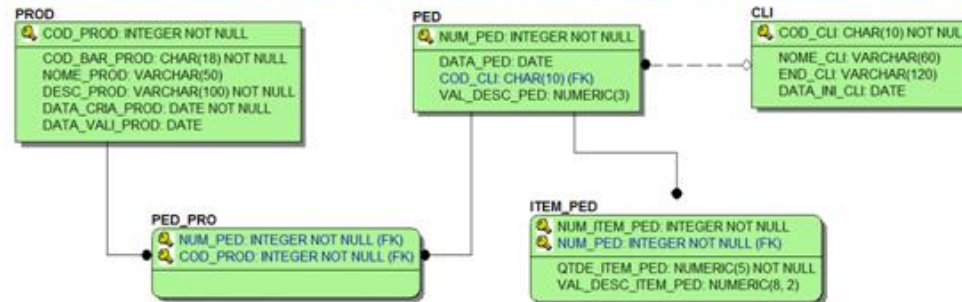
Conceitual



Lógica



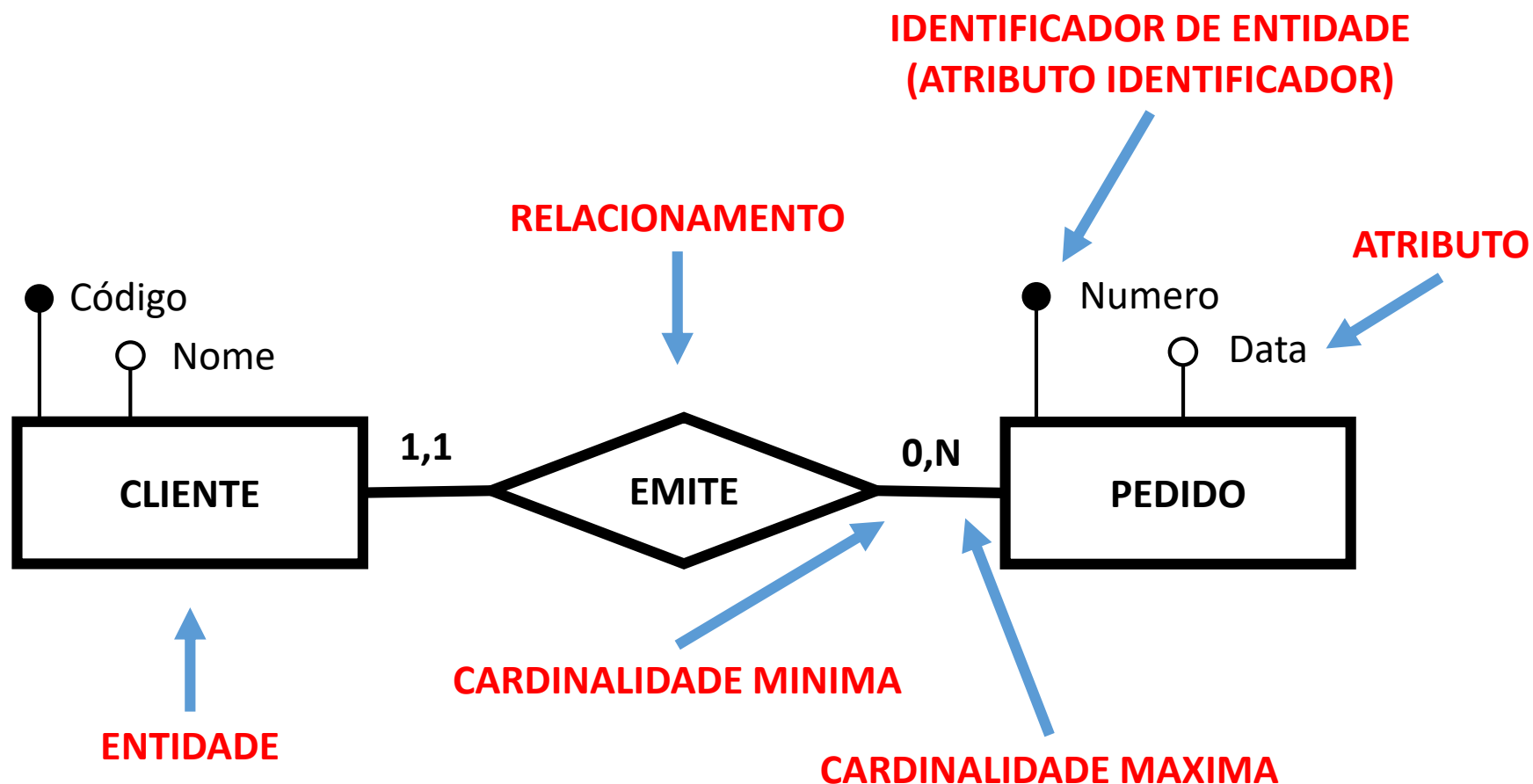
Física



Bancos de Dados

MODELAGEM DE DADOS

MODELO CONCEITUAL



Modelo Entidade e Relacionamento



Modelo Relacional Normalizado

1. Conceito

O **modelo lógico** têm seu início a partir do **Modelo Conceitual**, levando em consideração uma das três abordagens atualmente possíveis:

- Relacional,
- Hierárquica,
- Rede.

1. Conceito

O modelo lógico descreve as estruturas que estarão contidas no banco de dados, de acordo com as possibilidades permitidas pela abordagem, mas sem considerar, ainda, nenhuma característica específica de um SGBD, resultando em um esquema lógico de dados sob a ótica de uma das abordagens citadas.

1. Conceito

O **modelo lógico** apresenta uma **visão abstrata apropriada a equipe de desenvolvimento**.

Um **modelo lógico** eficiente tem que está normalizado e ter as chaves estrangeiras criadas corretamente.

1. Conceito

Observação

A partir deste ponto:

1) **tipo entidade** será citado como **tabela**;

2) **entidade** será citada como **registro**;

3) **atributo** será citado como **coluna** ou **campo**.

1. Conceito

A próxima etapa do projeto de banco de dados envolve o chamado **modelo relacional normalizado**.

Atualmente, grande parte dos sistemas de banco de dados utiliza o modelo relacional normalizado.

1. Conceito

Um banco de dados relacional é composto por *tabelas* **(também denominadas relações)**.

Observe a seguir alguns conceitos importantes para pleno entendimento do modelo relacional normalizado:

2. Tabelas

Estrutura bidimensional composta por linhas (tuplas) e **campos** (ou atributos).

The diagram illustrates a table structure with the following components labeled:

- Campo ou Atributo**: Points to the header row (CodDeppto, Nome).
- Nome da Tabela**: Points to the table title (Departamento).
- Linha ou Tupla**: Points to the first data row (D001, Financeiro).
- Coluna**: Points to the second column (Financeiro, Engenharia, Comercial).

Departamento	
CodDeppto	Nome
D001	Financeiro
D002	Engenharia
D003	Comercial

3. Chave Primaria (PK – Primary Key)

Atributo através do qual seja possível identificar determinado registro.

Uma **chave primária não pode ser repetida.**

3. Chave Primaria (PK – Primary Key)

O conjunto de valores que constituem a chave primária deve ser único dentro de uma tabela.

a) Chave primaria simples


b) Chave primaria composta

3. Chave Primaria (PK – Primary Key)

Chave primaria simples

Apenas um atributo (**campo**) compõe a chave primária. (**PK**)

Chave Primaria

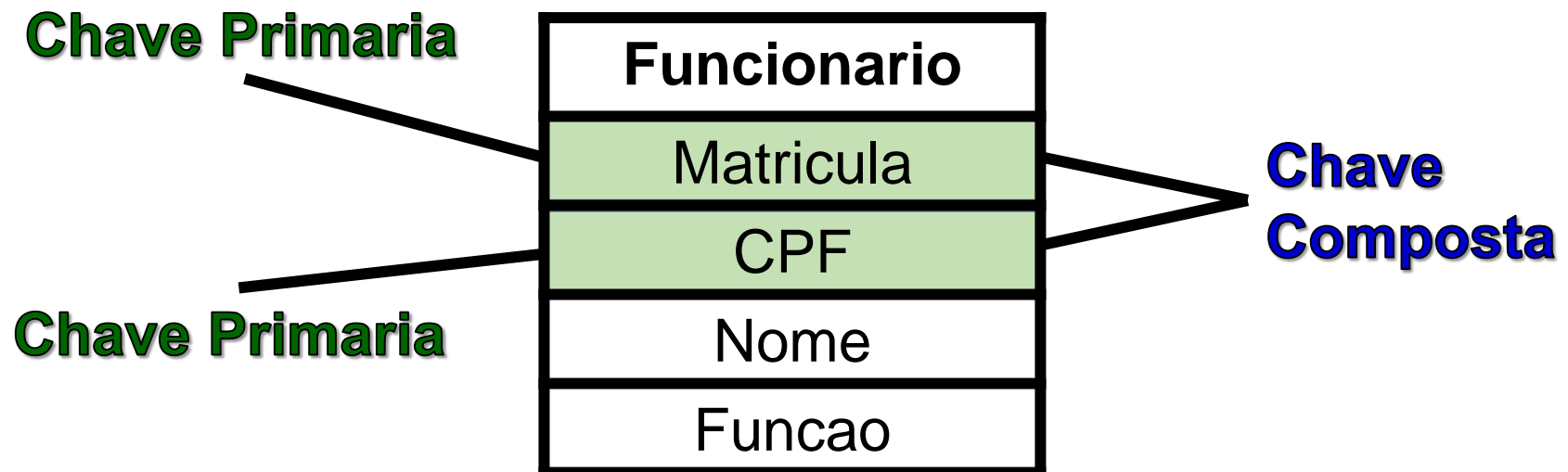


Departamento	
CodDeppto	Nome
D001	Financeiro
D002	Engenharia
D003	Comercial

3. Chave Primaria (PK – Primary Key)

Chave primaria composta

Mais de um atributo (**campo**) compõe a chave primária. (**PK**)



4. Chave Única (Unique)

Utilizada quando determinado campo não deve ser repetido e não é chave primária.

Aumenta a consistência do banco de dados.

4. Chave Única (Unique)

Exemplo:

Cadastro de funcionários recebe um código único que é a **chave primária**.

Para maior segurança e consistência podemos optar que o campo CPF também seja único, **evitando que o mesmo funcionário seja cadastrado duas vezes.**

MODELO LÓGICO

4. Chave Única (Unique)

Exemplo:

Chave Primaria

Chave Unica

FUNCIONARIO		
CodFunc	Nome	CPF
1001	Antonio	000111222-33
1002	Beatriz	111222333-44
1003	Claudio	222333444-55

5. Chave Estrangeira (FK – Foreign Key)

Utilizada **quando queremos que o valor de um atributo seja validado a partir do valor de atributo de outra tabela.**

Criamos assim uma relação de dependência (um relacionamento) entre as tabelas.

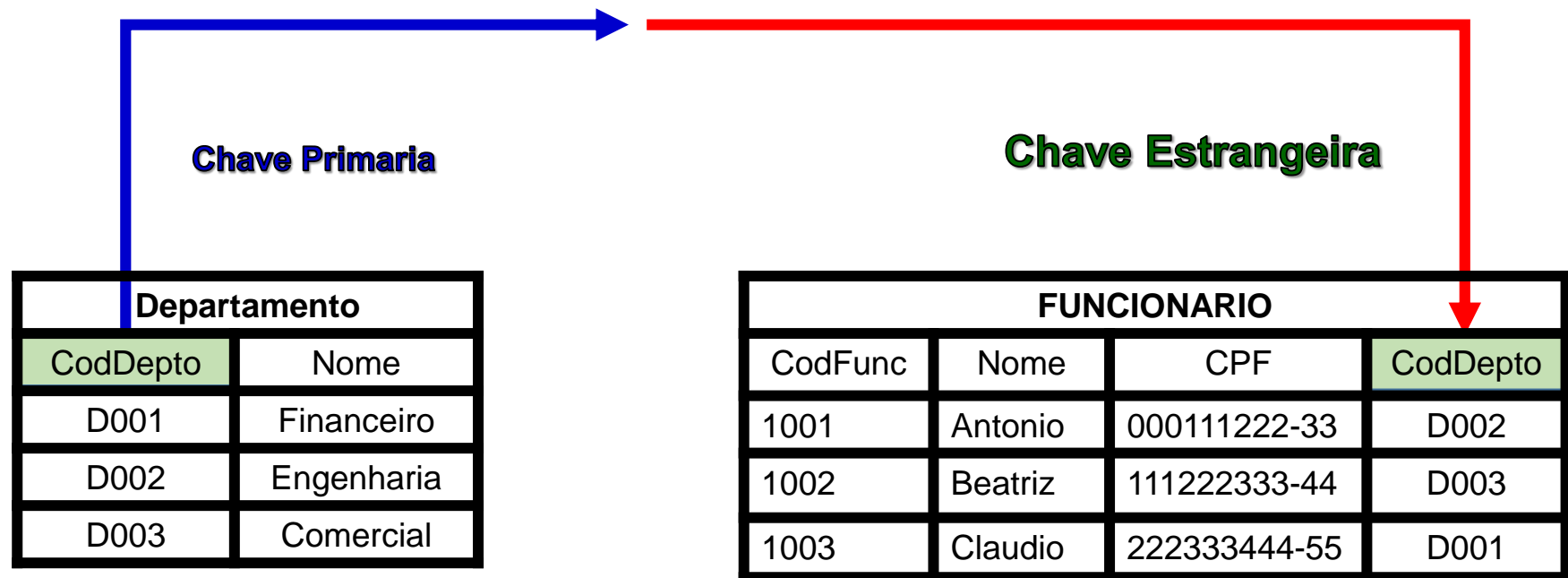
5. Chave Estrangeira (FK – Foreign Key)

Exemplo:

Antes de efetuar a alocação de um funcionário em um departamento, é necessário que o departamento em questão conste na tabela de departamento e na tabela funcionário.

5. Chave Estrangeira (FK – Foreign Key)

Exemplo:



RELACIONAMENTOS

6. Conceito Relacionamentos

Associação estabelecida **entre campos comuns de duas tabelas.**

Desta forma permitimos o estabelecimento de **correspondência entre registros de diferentes tabelas.**

Os relacionamentos apresentam a seguinte **classificação** quanto à sua **cardinalidade:**.

7. Tipos de Relacionamentos

Relacionamento um-para-um (1:1)

Relacionamento um-para-muitos (1:N)

Relacionamento muitos-para-muitos (N:N)

7. Tipos de Relacionamentos

Relacionamento um-para-um (1:1)

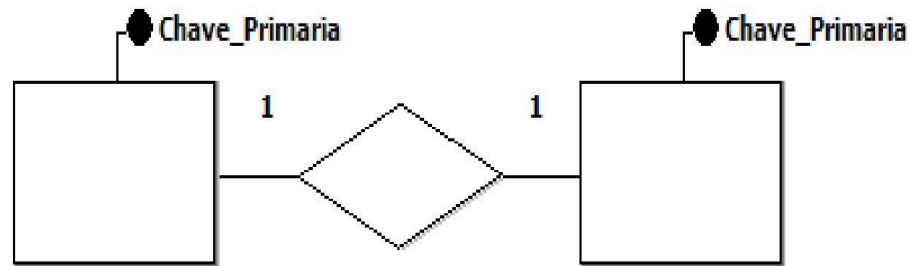
Cada ocorrência de uma tabela relaciona-se com uma e somente uma ocorrência da outra tabela.

Neste tipo de relacionamento, na maioria das vezes, a chave estrangeira pode ser criada em qualquer uma das tabelas.

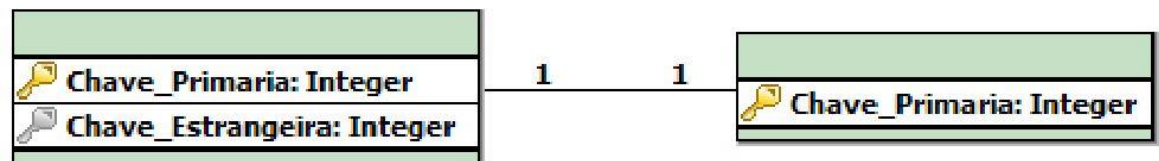
7. Tipos de Relacionamentos

Relacionamento um-para-um (1:1)

EXEMPLO



Modelo Conceitual

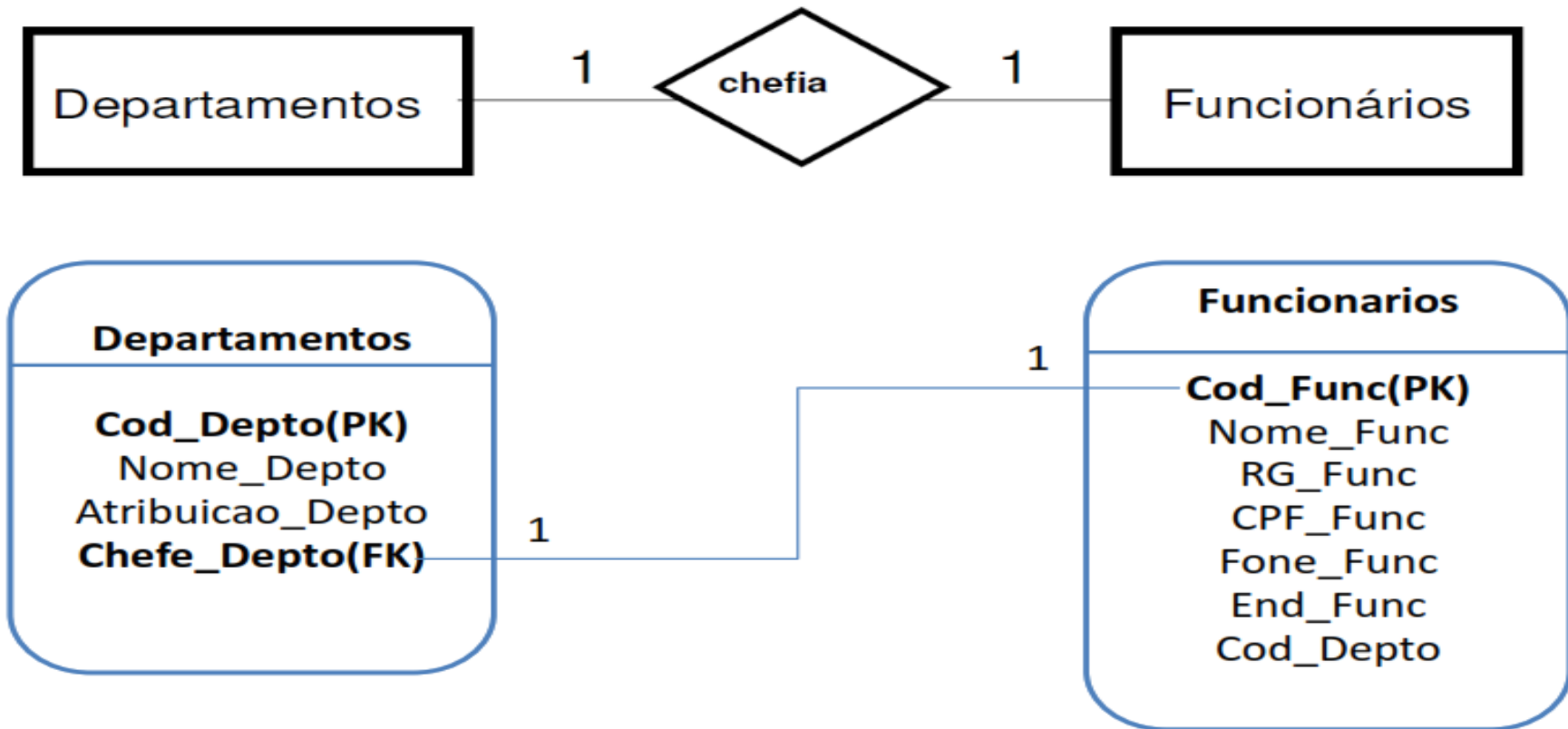


Modelo Lógico

7. Tipos de Relacionamentos

Relacionamento um-para-um (1:1)

EXEMPLO



7. Tipos de Relacionamentos

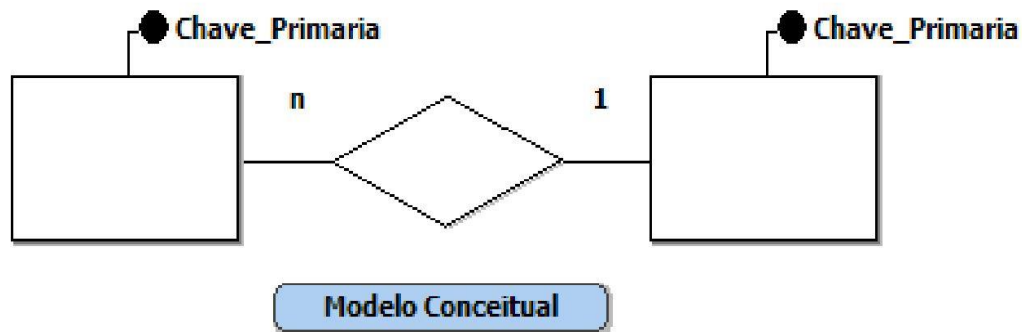
Relacionamento um-para-muitos (1:N)

Uma ocorrência da tabela pai relaciona-se com muitas ocorrências da tabela filho, mas **cada ocorrência da tabela filho somente pode estar relacionada com uma ocorrência da tabela pai**

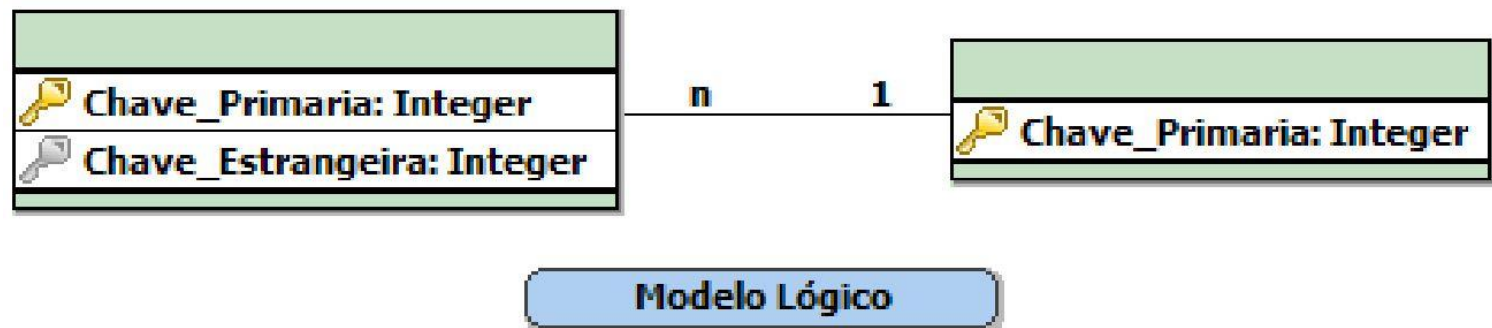
7. Tipos de Relacionamentos

Relacionamento um-para-muitos (1:N)

EXEMPLO



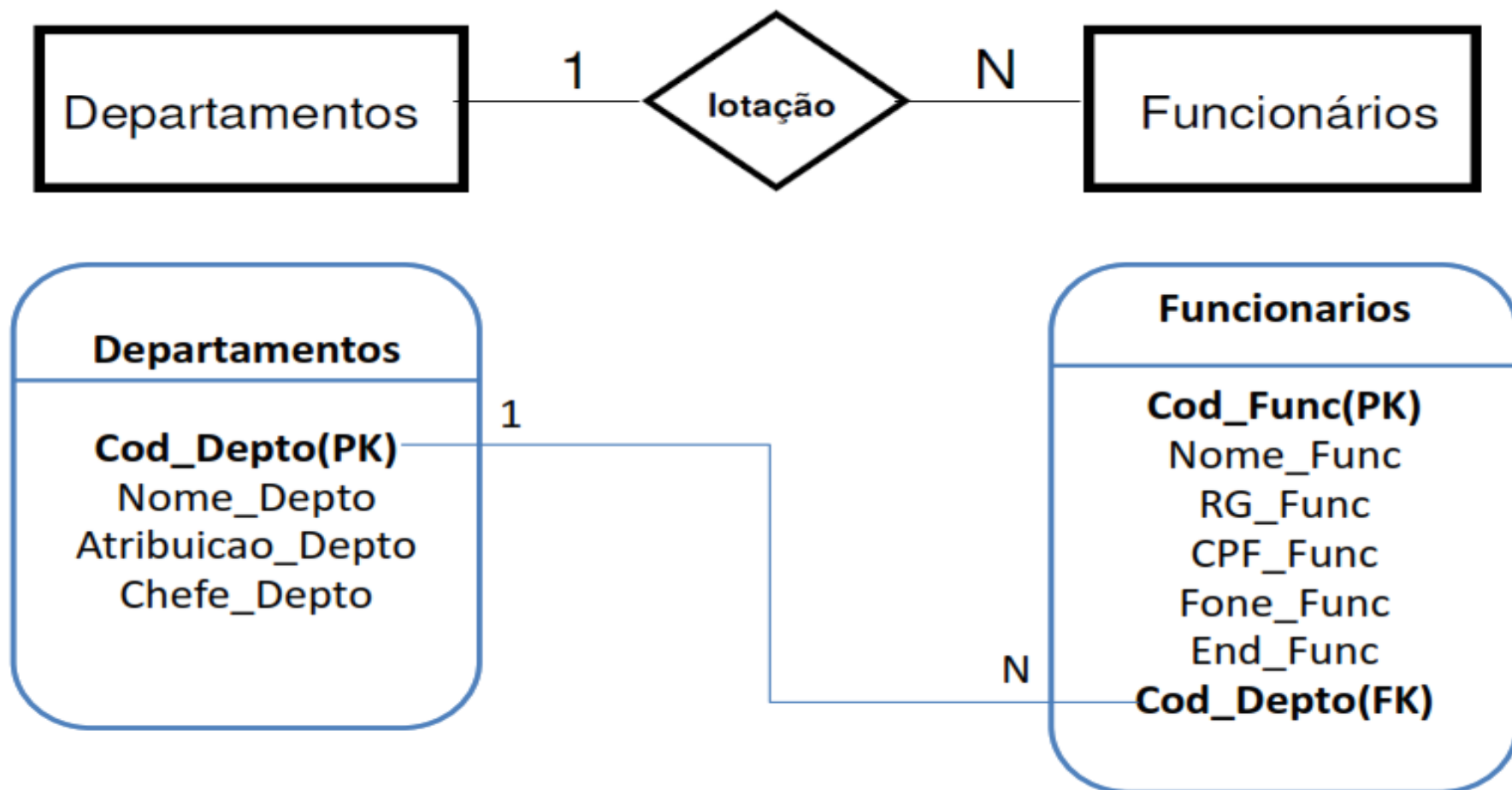
Neste tipo de relacionamento a **chave estrangeira** deve ser criada na tabela que tem a **cardinalidade N**



7. Tipos de Relacionamentos

Relacionamento um-para-muitos (1:N)

EXEMPLO



7. Tipos de Relacionamentos

Relacionamento muitos-para-muitos (N:N)

Apresenta em ambos os sentidos um ou mais relacionamentos de um-para-muitos.

No **modelo relacional** **não é possível** **efetuar este tipo de relacionamento de forma direta.**

Neste caso, deve-se **construir uma terceira tabela**(tabela de associação ou tabela de detalhes)

7. Tipos de Relacionamentos

Relacionamento muitos-para-muitos (N:N)

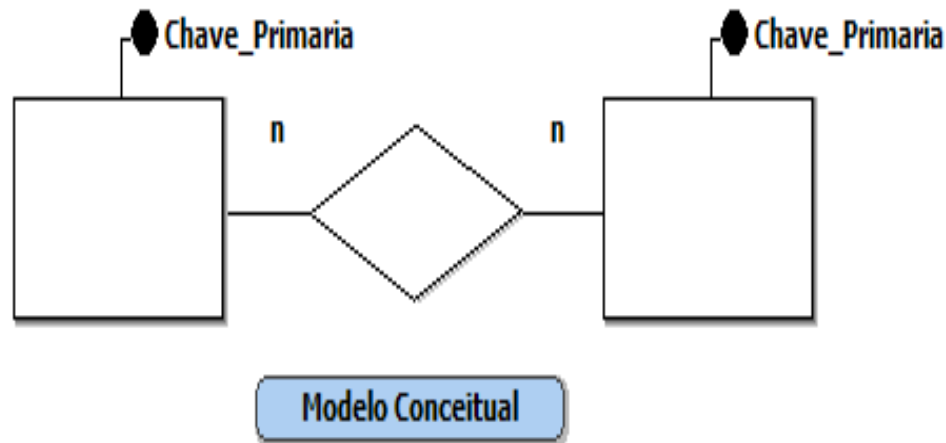
Essa tabela deve possuir chave primaria composta de dois campos e as chaves estrangeiras provenientes das duas tabelas originais.

Concluindo, **um relacionamento de muitos-para-muitos deve ser dividido em dois relacionamentos de um-para-muitos com uma terceira tabela.**

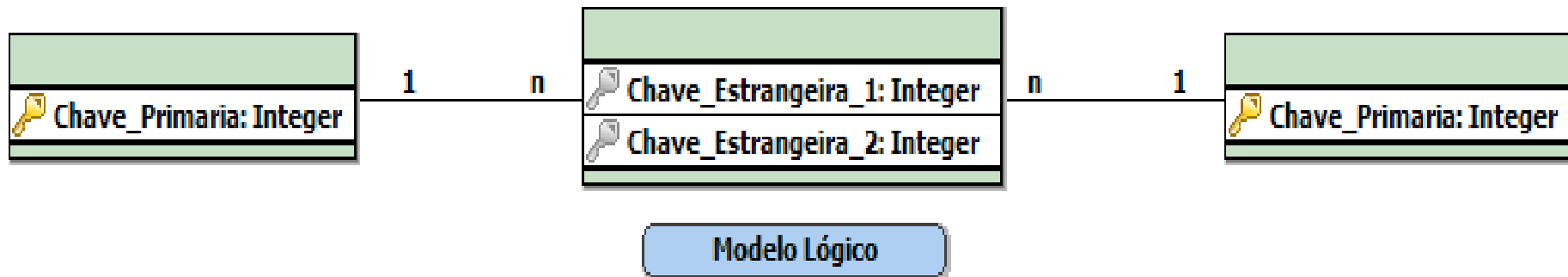
7. Tipos de Relacionamentos

Relacionamento muitos-para-muitos (N:N)

EXEMPLO



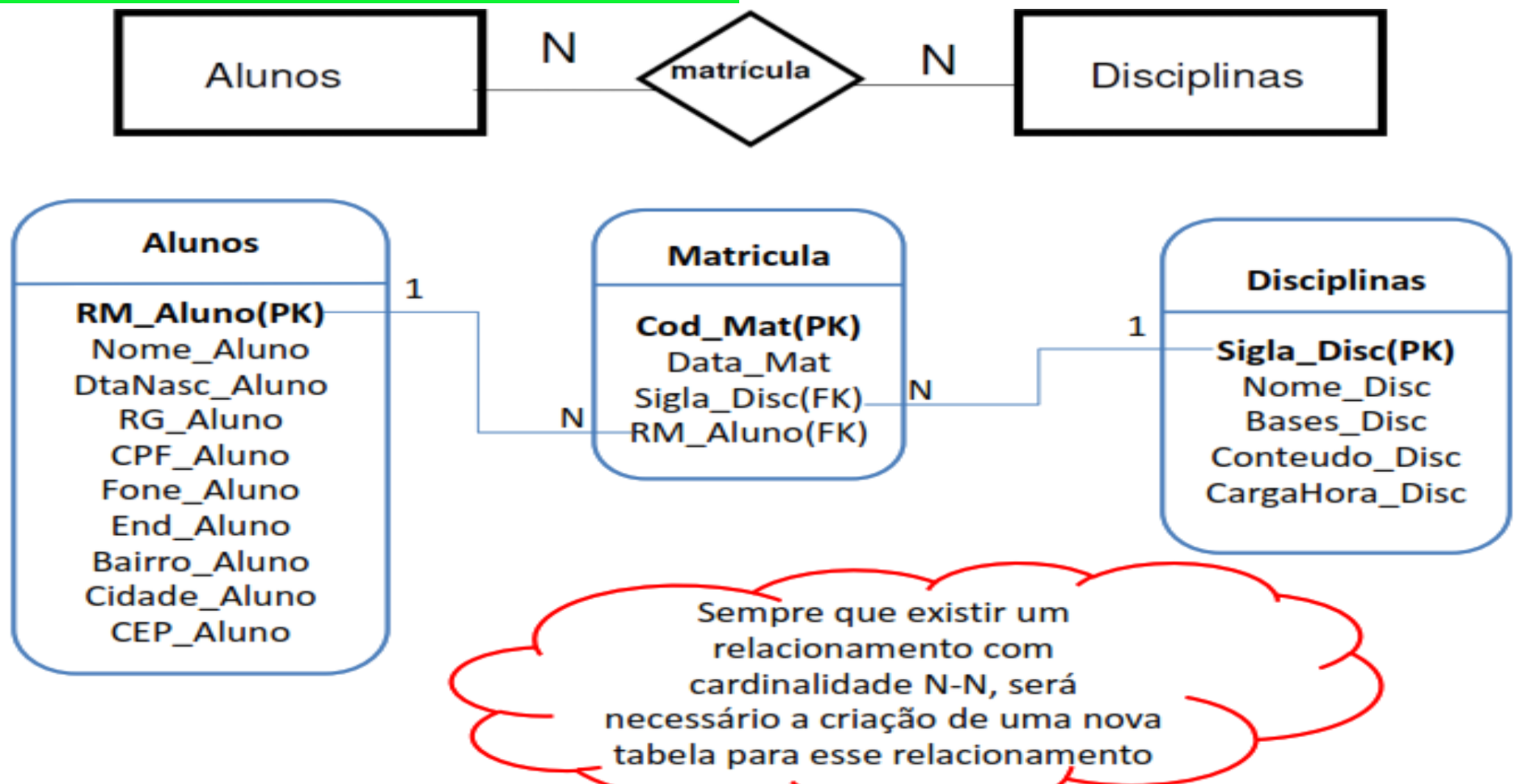
Neste tipo de relacionamento, deve-se criar uma nova tabela e inserir nela a chave estrangeira referente as tabelas envolvidas no relacionamento. As tabelas antigas passam a se relacionar através da nova tabela através de um cardinalidade 1:N



7. Tipos de Relacionamentos

Relacionamento muitos-para-muitos (N:N)

EXEMPLO



TIRINHA VIDA DE PROGRAMADOR

Ô CHEFE, EU TÔ MEXENDO NAS TABELAS DAQUELE SISTEMA QUE A EMPRESA JÁ TEM HÁ ALGUNS ANOS...

AH, SIM...
EU MESMO
QUE FIZ...

ENTÃO, VIM POR ISSO MESMO...
POR QUE É QUE TEM UMA TABELA
"NOMEEMPRESA" E UMA TABELA
"NOMEEMPRESA_NOMEEMPRESA"?

AH, É MEIO LÓGICO... A PRIMEIRA
GUARDA OS DADOS DA EMPRESA E
A SEGUNDA SÓ OS DADOS DO SITE
DA EMPRESA...

HMMM... TÁ... MAS E QUAL É A
DIFERENÇA ENTRE AS TABELAS
"FONTE" E "FONTES"?

NENHUMA.
UMA É BACKUP
DA OUTRA...

AI... BELEZA... MAS E QUAL É A
DIFERENÇA ENTRE AS TABELAS
"ATIVIDADE", "ATIVIDADES" E
"ATIVIDADESS"?

AH, AÍ É SÓ VER O NÚMERO DE
"S". UMA É A DE ATIVIDADES DO
DIA, A OUTRA DA SEMANA E A
OUTRA DO MÊS...

PLOFT!

**VIDA DE
PROGRAMADOR**

real historia;
string sender = "Assembly";

#2037



NOTAÇÃO RESUMIDA

8. Notação Resumida

Modelos Lógicos Relacionais)

Notação compacta, útil para discussões sobre a estrutura geral do banco de dados, utilizada quando não se deseja entrar no nível maior de detalhamento.

Observe o exemplo a seguir:

8. Notação Resumida

Modelos Lógicos Relacionais)

Departamento (CodDept, Nome)

Funcionario (CodFun, Nome, CPF, CodDept)

➤ **CodDept referencia Departamento.**

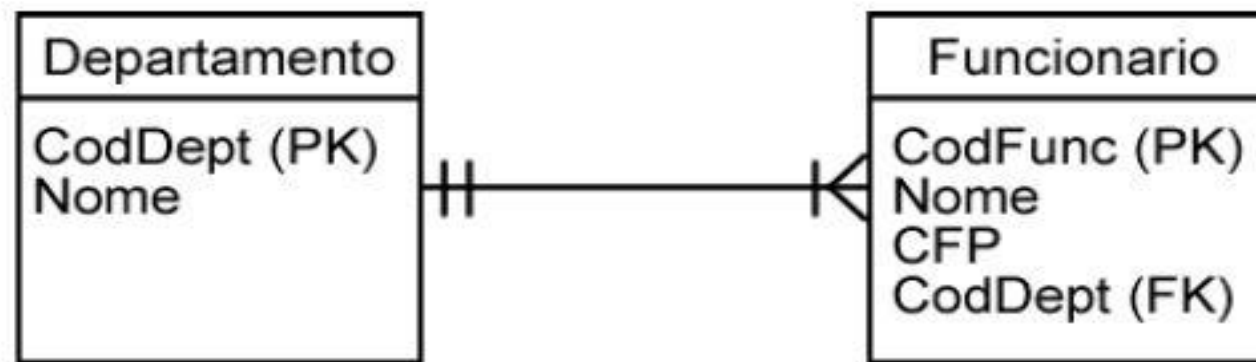
A notação resumida acima representa o seguinte relacionamento entre as tabelas Departamento e Funcionario.

8. Notação Resumida

Modelos Lógicos Relacionais)

Departamento (CodDept, Nome)

Funcionario (CodFun, Nome, CPF, CodDept)



Relacionamento **Departamento** x **Funcionario**

INTEGRIDADE

9. Integridade de Dados

Impor a **integridade de dados** garante a **qualidade dos dados** em um banco de dados.

Os dados devem **refletir corretamente a realidade** *representada pelo banco e também devem ser consistentes entre si.*

10. Integridade de Domínio

Zela pelos valores ideais e necessários para um atributo.

Para isso definimos algumas regras de validação por meio de expressões compostas de valores constantes.

10. Integridade de Domínio

Exemplos

- ✓ Não permitir um estoque negativo;
- ✓ Impedir uma data de nascimento superior à data atual;
- ✓ Não permitir que o valor de um produto seja negativo.

11. Integridade de Entidade

Tem o objetivo de validar os valores permitidos a partir de valores já inseridos na própria entidade.

Após uma “auto-consulta” a entidade vai permitir ou não a gravação do novo registro.

11. Integridade de Entidade

Exemplos

- ✓ Não permitir duas pessoas com o mesmo CPF;
- ✓ Impedir a locação de uma fita que já está locada.

12. Integridade Referencial

Zela pela consistência dos registros de uma entidade a partir de valores provenientes de outras entidades, isto é, determinado registro vai “depende” diretamente de um registro de outra tabela.

12. Integridade Referencial

Exemplos

- Um registro em uma tabela pai pode ter **um** ou **mais registros** em uma tabela filho.

12. Integridade Referencial

Exemplos

- Um registro em uma tabela filho sempre ter **um registros** coincidente em uma tabela pai.

12. Integridade Referencial

Exemplos

- Para a inclusão de um registro em uma determinada **tabela filho**, é necessário que exista um registro **pai** **coincidente**.

12. Integridade Referencial

Exemplos

- Um registro pai só poderá ser excluído se não possuir nenhum registro filho.

PRATICANDO o APRENDIDO

**Aluno deve responder
Questionário elaborado no
FORMs**

**009 – Modelo Lógico
Parte I**

MINI MUNDO

Entrevista o usuário do B.D.
Requisitos Funcionais do banco de dados

Levantamento e
Análise das
necessidades

Esquema Conceitual
(Diagrama de Entidade e Relacionamento)
Modelo de Entidade-Relacionamento

Projeto
Conceitual

Esquema Lógico
(Modelo Relacional Normalizado)
Descreve as estruturas que estarão contidas no B.D

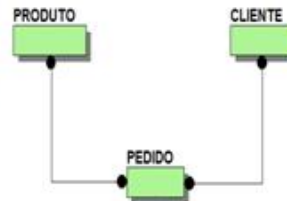
Projeto Lógico
do Banco de
Dados

Especificação de Transações e rotinas
(Dicionário de Dados)

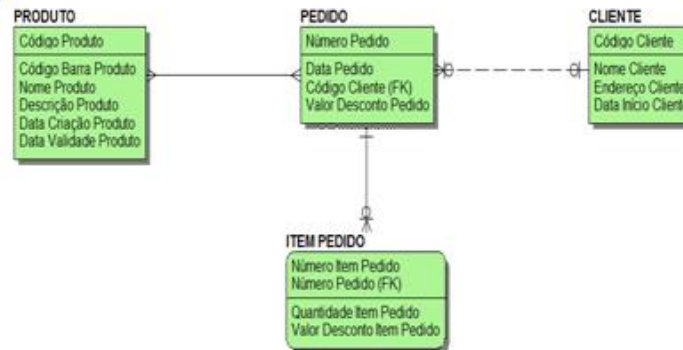
Projeto Físico
do Banco de
Dados

NEGÓCIO

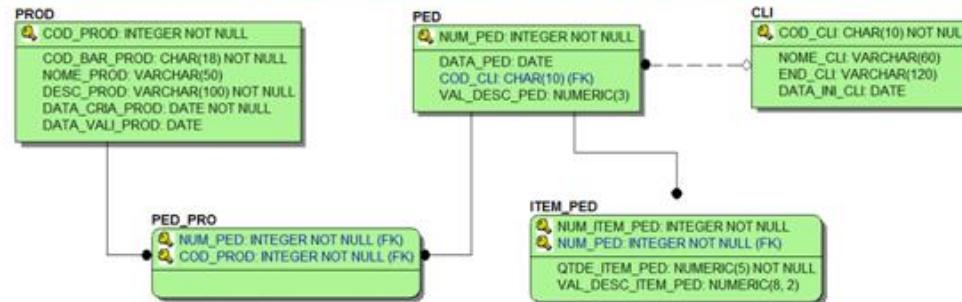
Conceitual



Lógica



Física



Bancos de Dados

MODELAGEM DE DADOS

13. REFERENCIAS

Slide Projeto Conceitual de B.D - Crysthiane Carvalho
paola@spei.br

Apostila Modelagem de Banco de Dados – Prof. Marcos Alexandruk

Informática – Banco de Dados ; FRANÇA - Cicero T. P. Lima ; JUNIOR - Joaquin Celestino; Editora UAB/UECE -- Fortaleza – 2014 ,

Sistemas de banco de dados / Ramez Elmasri e Shamkant B. Navathe ; tradução Daniel Vieira ; 6ª. ed. – São Paulo : Pearson Addison Wesley, 2011.

FIM