

# Banco de Dados

Modelo Relacional

Professor: Paulo Sérgio Ruiz Del Aguila

## Projeto do Banco de Dados

- Projeto Conceitual
  - Modelo Entidade Relacionamento
  - Entrada: Minimundo
  - Saída: Esquema Conceitual
- Projeto Lógico
  - Modelo Relacional
  - Entrada: Esquema Conceitual
  - Saída: Esquema Lógico
- Projeto Físico
  - Modelo SQL DDL do SGBD a ser usado
  - Entrada: Esquema Lógico
  - Saída: Esquema Físico SQL DDL do SGBD usado

## Modelo Relacional

- Foi definido em 1970 por E.F. Codd.
- Principais razões para sua grande aceitação:
  - Simplicidade (teoria dos conjuntos) e Formalismo (álgebra relacional).
- É um modelo lógico que representa dados como relações.
  - Neste modelo, o BD é visto como um conjunto de tabelas.
  - Os dados em uma tabela representam fatos reais a respeito de uma entidade ou de um relacionamento do mundo real.

3

## Modelo Relacional

- Composição básica de um BD Relacional
  - Tabelas (ou Relações)
    - Compostas de:
      - Linhas
      - Colunas
      - Chaves primárias
    - Relacionadas através de:
      - Chaves estrangeiras

4

## Modelo Relacional

- Propriedades de uma Tabela (Relação):
  - Não há tuplas duplicadas em uma mesma tabela.
  - As tuplas de uma tabela não são ordenadas.
  - Os atributos em uma tupla não são ordenados.
  - Cada tabela possui um número fixo de atributos, todos com nomes distintos.

5

## Modelo Relacional

- Terminologias
  - Tabela = Relação = Arquivo.
  - Linha = Tupla = Registro.
  - Coluna = Atributo = Campo.
  - Valor de Coluna = Valor de Atributo = Valor de Campo.

6

## Modelo Relacional

- Chaves
  - Conceito usado para especificar restrições de integridade básicas de um SGBD Relacional.
  - Quatro tipos:
    - Chave candidata (ex. matrícula e cpf).
    - Chave primária (ex. matrícula).
    - Chave alternativa (ex. cpf).
    - Chave estrangeira (ex. cepEnd).

Endereço			
cep <sup>PK</sup>	rua	bairro	cidade

Cliente			
matricula <sup>PK</sup>	cpf	nome	cepEnd <sup>FK</sup>

7

## Modelo Relacional

- Chaves
  - Uma chave primária é representada pela sigla **PK** (Primary Key).
  - Uma chave estrangeira é representada pela sigla **FK** (Foreign Key).

8

## Modelo Relacional

- Restrições de integridade básicas
  - Integridade de domínio.
  - Integridade de chave.
  - Integridade de vazio.
  - Integridade referencial.
- As restrições acima são:
  - Garantidas automaticamente por um SGBD relacional.
  - Não sendo exigido que o programador escreva procedimentos para garanti-las explicitamente.

9

## Modelo Relacional

- Restrições de integridade básicas
  - Integridade de domínio
    - Especifica que para uma coluna A de uma tabela, todo valor associado a A deve ser atômico e pertencer ao domínio desta coluna.
  - Integridade de chave
    - Especifica que os valores das chaves primárias devem ser únicos (não pode haver tuplas duplicadas em uma tabela) e não vazios. (UNIQUE e NOT NULL)

10

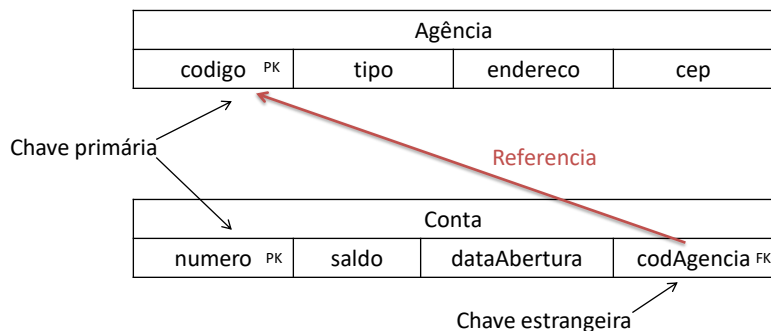
# Modelo Relacional

- Restrições de integridade básicas
  - Integridade de vazio
    - Controla quais colunas de uma tabela podem receber valores nulos. (**NULL ou NOT NULL**)
  - Integridade referencial
    - Especifica que os valores de uma chave estrangeira devem aparecer na chave primária da tabela referenciada. (**FOREIGN KEY REFERENCES**)

11

# Modelo Relacional

- Esquema lógico = definição das tabelas
  - Representação básica



12

## Modelo Relacional

- Esquema lógico = definição das tabelas
  - Outra representação

Agencia (codigo, tipo, endereço, cep)

Conta (numero, saldo, dataAbertura, **codAgencia**)  
codAgencia referencia Agencia

13

## Transformação ER/Relacional

- As regras foram definidas tendo por base, entre outros, os seguintes princípios :
  - Evitar junções.
  - Diminuir o número de chaves.
  - Evitar campos opcionais.

14

## Transformação ER/Relacional

- Evitar junções
  - Junção é uma operação que busca dados de diversas linhas através da igualdade de campos.
  - Junções envolvem comparações entre diversas linhas.
    - Junções requerem diversos acessos a disco (minimizam o desempenho).

LISTAR O CÓDIGO DA AGENCIA E O NUMERO E SALDO DA CONTA:  
`SELECT A.CODIGO, C.NUMERO, C.SALDO FROM AGENCIA A, CONTA C  
WHERE C.CODAGENCIA = A.CODIGO;`

15

## Transformação ER/Relacional

- Evitar junções
  - Assim, quando possível, evite usar junções.
  - Isso significa, ter os dados necessários ao resultado da consulta em uma única linha da tabela.

16



## Transformação ER/Relacional

- Diminuir o número de chaves
  - Para implementar eficientemente o controle de chaves primárias o SGBD usa índices para cada chave primária.
    - Índices tendem a ocupar espaço considerável em disco.
    - A Inserção ou remoção de entradas em um índice podem exigir diversos acesso a disco.

17

## Transformação ER/Relacional

- Diminuir o número de chaves
  - Assim, quando possível, diminua a quantidade de chaves primárias.
  - Isso significa ter os dados subordinados as chaves primárias em uma única tabela.

18

## Transformação ER/Relacional

- Diminuir o número de chaves

– Exemplo:

- Este é melhor pois cria-se um único índice

Cliente					
cpf <sup>PK</sup>	nome	nomeContato	endereço	email	telefone

- Neste há dois índices com exatamente as mesmas entradas

Cliente	
cpf <sup>PK</sup>	nome

ClienteContato				
cpfCli <sup>PK</sup> <sub>FK</sub>	nomeContato	endereço	email	telefone

19

## Transformação ER/Relacional

- Evitar campos opcionais
  - Atributos opcionais devem ser tratados via programação, então há mais trabalho quando temos campos opcionais em grande número.
  - Para evitá-los devemos usar a regra de transformação de um relacionamento que gere nenhum ou menos atributos opcionais.

20

## Regras Transformação ER/Relacional

- Passos da transformação ER para Relacional
  - Tradução inicial de entidades e respectivos atributos.
  - Tradução de relacionamentos e respectivos atributos.
  - Tradução de generalizações/ especializações.

21

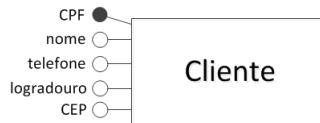
## Regras Transformação ER/Relacional

- Implementação inicial de entidades
  - Cada entidade é traduzida para uma tabela.
  - Cada atributo da entidade define uma coluna desta tabela.
  - Atributos identificadores da entidade correspondem a chave primária da tabela.

22

## Regras Transformação ER/Relacional

- Implementação inicial de entidades



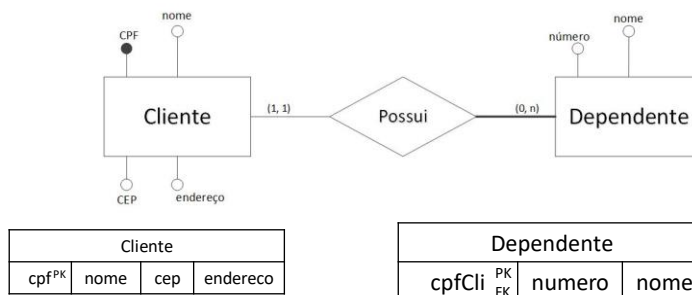
Cliente				
cpf <sup>PK</sup>	nome	telefone	logradouro	cep

Cliente (cpf, nome, telefone, logradouro, cep)

23

## Regras Transformação ER/Relacional

- Tradução de entidade fraca



- Faz-se a tradução das entidades fracas em tabelas, seus atributos em campos e adiciona-se a PK da entidade forte na PK da entidade fraca.

24

## Regras Transformação ER/Relacional

- Nomes de colunas
  - Use nomes das colunas que sejam curtos e significativos.
  - Nos SGBDR os nomes das colunas não podem conter brancos.
  - Nomes de colunas não necessitam conter o nome da tabela.
    - Ex. Prefira usar nome do que usar nomeCli, nomeFor, etc..

25

## Regras Transformação ER/Relacional

- Nome da coluna da chave primária
  - A chave primária pode aparecer em outras tabelas na forma de **chave estrangeira**.
  - Nomes das colunas que compõem a **chave estrangeira** devem vir sufixadas ou prefixadas com o nome ou sigla da tabela que pertencem como chave primária.
    - Ex. codigoCli, codigoFor, codigoProd, etc..

Chave estrangeira que referencia o código do Cliente

26

## Regras Transformação ER/Relacional

- Implementação de relacionamento
  - Existem diferentes estratégias para a transformação de relacionamentos no modelo lógico:
    - Tabela própria.
    - Adição de colunas a uma das tabelas.
    - Fusão de tabelas.
  - A escolha de uma dessas alternativas depende da cardinalidade (máxima e mínima do relacionamento).

27

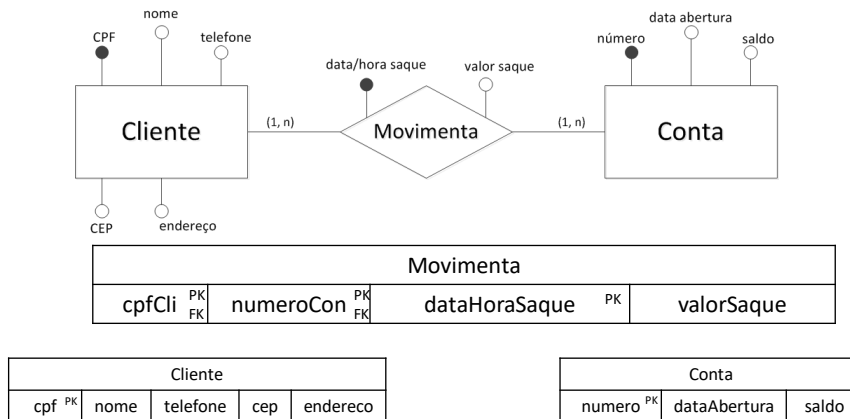
## Regras Transformação ER/Relacional

- Tabela própria
  - Nesta, o relacionamento é implementado através de uma tabela própria, que contém os atributos identificadores das entidades participantes + os atributos do relacionamento.
  - Ocorre quando o relacionamento tem cardinalidade N:N.

28

## Regras Transformação ER/Relacional

- Tabela própria



29

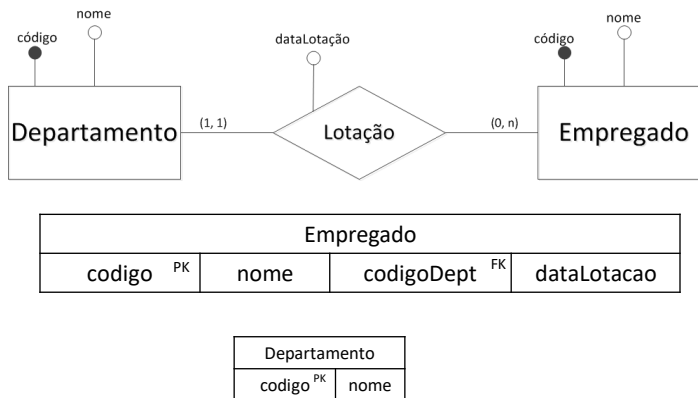
## Regras Transformação ER/Relacional

- Adição de colunas
  - Nesta, o relacionamento é implementado através da inserção de colunas na tabela oposta a multiplicidade máxima 1, ou seja, do lado N.
  - Ocorre quando o relacionamento tem cardinalidade N:1 ou 1:N.

30

## Regras Transformação ER/Relacional

- Adição de colunas



31

## Regras Transformação ER/Relacional

- Fusão de tabelas
  - Nesta, o relacionamento é implementado através da união das tabelas participantes.
  - Pode ocorrer quando o relacionamento é obrigatório (mínimas 1) e tem cardinalidade 1:1

32



## Regras Transformação ER/Relacional

- Fusão de tabelas
  - Relacionamento obrigatório com cardinalidade 1:1



Confer��ncia			
codigo <sup>PK</sup>	nome	dataOrg	endereçoCom

33

## Regras Transforma  o ER/Relacional

- Relacionamentos 1:1

Cardinalidade	Regra de implementa��o		
	Tabela pr��pria	Adi��o de coluna	Fus��o de tabelas
(0,1) (0,1)	2� op��o	✓	✗
(0,1) (1,1)	3� op��o	2� op��o	✓
(1,1) (1,1)	3� op��o	2� op��o	✓

34

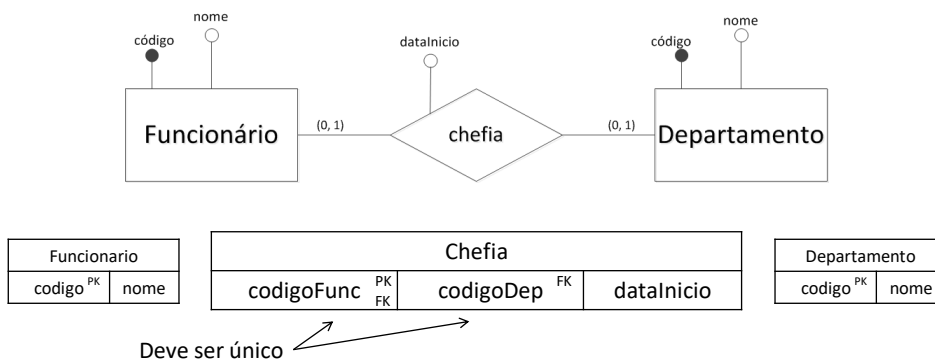
## Regras Transformação ER/Relacional

- Relacionamentos 1:1 onde ambas entidades têm relacionamentos opcionais.
  - Cria-se a tabela de relacionamento e escolhe-se uma das chaves das entidades arbitrariamente para ser a chave primária e estrangeira da tabela do relacionamento.

35

## Regras Transformação ER/Relacional

- Relacionamentos 1:1 onde ambas entidades têm relacionamentos opcionais.
  - Tabela própria



36

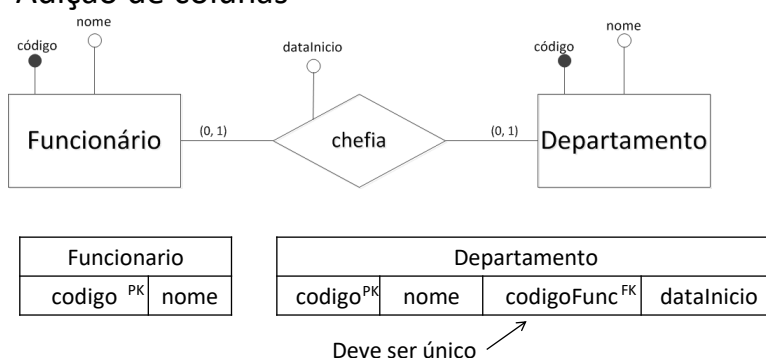
## Regras Transformação ER/Relacional

- Relacionamentos 1:1 onde ambas entidades têm relacionamentos opcionais.
  - Pode-se usar adição de colunas.
  - Escolhe-se um dos lados arbitrariamente para receber a chave estrangeira.

37

## Regras Transformação ER/Relacional

- Relacionamentos 1:1 onde ambas entidades têm relacionamentos opcionais.
  - Adição de colunas



38

## Regras Transformação ER/Relacional

- Relacionamentos 1:1 onde ambas entidades tem relacionamentos opcionais
  - A solução por adição de colunas é a melhor, pois minimiza a quantidade de junções e chaves, entretanto, pode-se ter atributos opcionais, os quais devem ser tratados via programação.
  - A solução de tabela própria é aceitável, mas a de fusão de tabela é semanticamente inviável para o contexto.

39

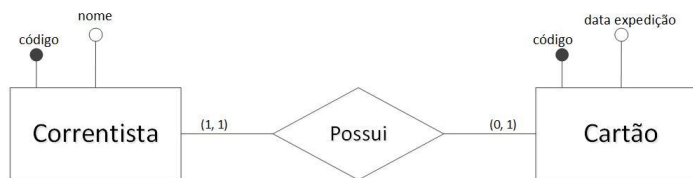
## Regras Transformação ER/Relacional

- Relacionamentos 1:1 onde uma das entidades tem relacionamento opcional.
  - O relacionamento pode ser implementado através da união das tabelas participantes.
  - Pode-se também usar adição de colunas, onde o lado da entidade de relacionamento opcional recebe a chave estrangeira.

40

## Regras Transformação ER/Relacional

- Relacionamentos 1:1 onde uma das entidades tem relacionamento opcional.
  - Fusão de tabelas

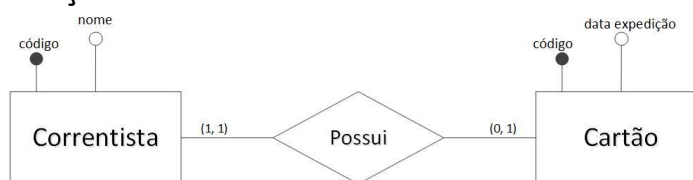


Correntista			
codigo <sup>PK</sup>	nome	codigoCartao	dataExpedicao

41

## Regras Transformação ER/Relacional

- Relacionamentos 1:1 onde uma das entidades tem relacionamento opcional.
  - Adição de coluna



Correntista	
codigo <sup>PK</sup>	nome

Cartão		
codigo <sup>PK</sup>	dataExpedicao	codigoCor <sup>FK</sup>

Deve ser único →

42

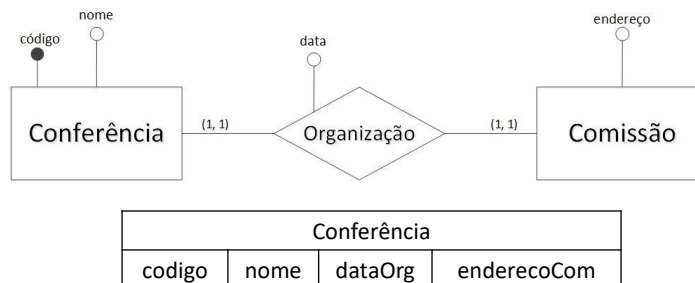
## Regras Transformação ER/Relacional

- Relacionamentos 1:1 onde uma das entidades tem relacionamento opcional.
  - A solução fusão de tabelas é a melhor, pois minimiza a quantidade de junções e chaves, entretanto, pode-se ter atributos opcionais, os quais devem ser tratados via programação.
  - A solução de adição de coluna também é possível, mas aumentaria a quantidade de junções e chaves. Neste caso a solução de tabela própria seria a pior opção.

43

## Regras Transformação ER/Relacional

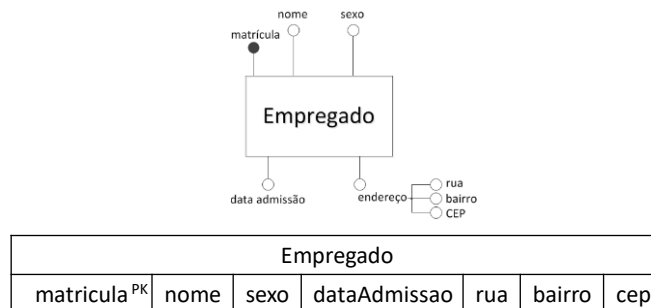
- Relacionamentos 1:1 onde temos relacionamento obrigatório
  - Usa-se fusão de tabelas.



44

## Regras Transformação ER/Relacional

- Atributos compostos
  - Somente os atributos simples que compõem o atributo composto são incluídos na nova tabela.



45

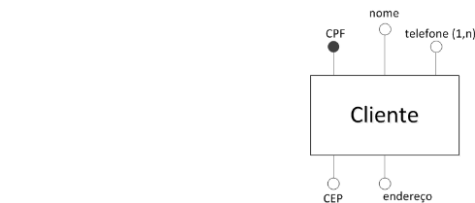
## Regras Transformação ER/Relacional

- Atributos multivalorados
  - Duas novas tabelas são criadas.
  - A primeira tabela contém todos os atributos da entidade, exceto o atributo multivalorado.
  - A segunda tabela refere-se ao atributo multivalorado.
  - Dois atributos formam a chave da segunda tabela.

46

## Regras Transformação ER/Relacional

- Atributos multivalorados



Cliente			
cpf <sup>PK</sup>	nome	endereço	cep

Telefone	
cpfCli <sup>PK</sup> <sub>FK</sub>	numTelefone <sup>PK</sup>

Outra abordagem:

Telefone		
idTelefone <sup>PK</sup>	numero	cpfCli <sup>FK</sup>

47

## Regras Transformação ER/Relacional

- Entidades associativas
  - Cria-se três tabelas
    - Uma para cada entidade participante.
    - Uma para a entidade associativa.
  - Caso um identificador não tenha sido atribuído a entidade associativa.
    - A chave primária da entidade associativa consiste das chaves primárias das entidades participantes.

48



## Regras Transformação ER/Relacional

- Entidades associativas



Ordem	
id <sup>PK</sup>	data

Produto		
codigo <sup>PK</sup>	descricao	preco

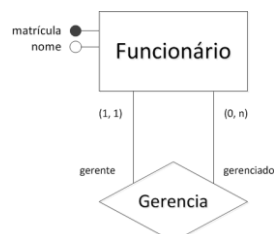
Lista			
idOrdem <sup>PK</sup> <sub>FK</sub>	codProduto <sup>PK</sup> <sub>FK</sub>	quantidade	

49

## Regras Transformação ER/Relacional

- Relacionamentos unários (1:N)

- Uma chave estrangeira é acrescentada dentro da mesma relação que referencia os valores da chave primária.



Funcionario		
matricula <sup>PK</sup>	nome	matriculaGerente <sup>FK</sup>

50

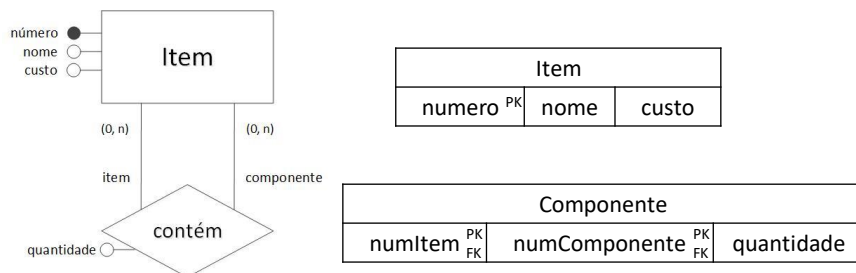
## Regras Transformação ER/Relacional

- Relacionamentos unários (N:N)
  - Duas tabelas são criadas.
  - Uma representa a entidade participante do relacionamento.
  - A outra representa o relacionamento N:N.

51

## Regras Transformação ER/Relacional

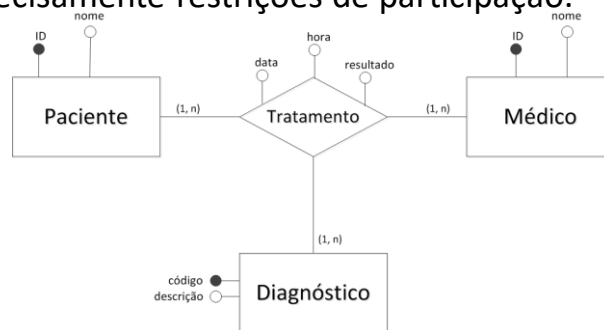
- Relacionamentos unários (N:N)



52

## Regras Transformação ER/Relacional

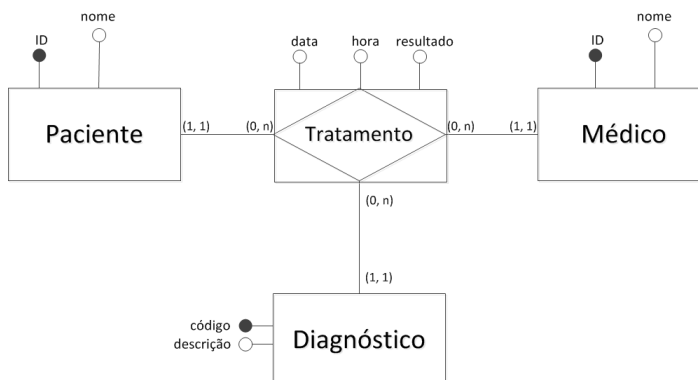
- Relacionamentos ternários
  - Converte o relacionamento ternário em uma entidade associativa de forma a representar mais precisamente restrições de participação.



53

## Regras Transformação ER/Relacional

- Relacionamentos ternários
  - relacionamento ternário → entidade associativa



54

## Regras Transformação ER/Relacional

- Relacionamentos ternários

Paciente	
id <sup>PK</sup>	nome

Medico	
id <sup>PK</sup>	nome

Diagnóstico	
codigo <sup>PK</sup>	descricao

Tratamento					
idPaciente <sup>PK</sup> <sub>FK</sub>	idMedico <sup>PK</sup> <sub>FK</sub>	codDiagnostico <sup>PK</sup> <sub>FK</sub>	data	hora	resultado

55

## Regras Transformação ER/Relacional

- Generalização/Especialização

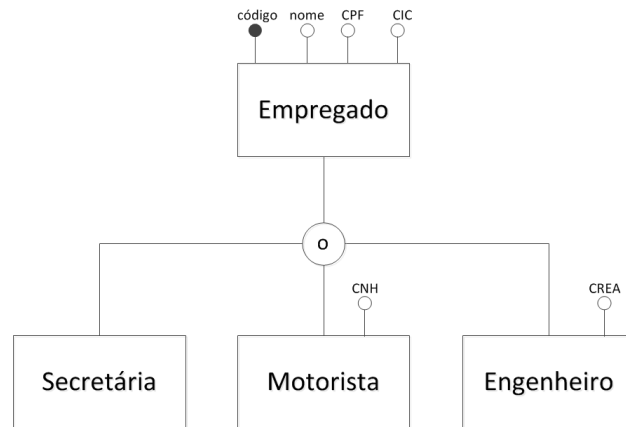
– Duas alternativas básicas

- Uso de uma única tabela para toda a hierarquia.
- Uso de uma tabela para cada entidade, desde que essa entidade tenha atributos próprios.

56

## Regras Transformação ER/Relacional

- Generalização/Especialização



57

## Regras Transformação ER/Relacional

- Generalização/Especialização

- Uso de uma única tabela para toda a hierarquia.
  - Todas as sub-classes são fundidas em uma única tabela, que contém:
    - Os atributos da superclasse.
    - Os atributos das subclasses, que serão opcionais.
    - Caso não exista uma coluna tipo, a mesma deve ser adicionada.

58

## Regras Transformação ER/Relacional

- Generalização/Especialização
  - Uso de uma única tabela para toda a hierarquia.

Empregado						
codigo <sup>PK</sup>	nome	cpf	cic	tipo	cnh	crea

59

## Regras Transformação ER/Relacional

- Generalização/Especialização
  - Uso de uma tabela para cada entidade.
    - Cria-se uma tabela para cada entidade que compõe a hierarquia, \*desde que essa entidade tenha atributos próprios.
      - Uma tabela para a superclasse, contendo seus atributos e uma coluna tipo.
      - Uma tabela para cada subclasse\*, contendo seus atributos e a PK da superclasse.

60

## Regras Transformação ER/Relacional

- Generalização/Especialização
  - Uso de uma tabela para cada entidade.

Empregado				
codigo <sup>PK</sup>	nome	cpf	cic	tipo

Motorista	
codEmp <sup>PK</sup> <sub>FK</sub>	cnh

Engenheiro	
codEmp <sup>PK</sup> <sub>FK</sub>	crea

61

## Regras Transformação ER/Relacional

- Generalização/Especialização
  - Discussão:
    - O uso de uma única tabela para toda hierarquia minimiza junções e diminui a quantidade de chaves, entretanto, tem-se atributos opcionais, os quais devem ser tratados por programação.
    - A solução com uso de uma tabela para cada entidade não tem atributos opcionais, mas apresenta um número maior de junções e chaves.

62

# Obrigado

E-mail: paulo.aguila@ifam.edu.br  
Telefone: (92) 98189-8899